3D/2D modelling suite for integral water solutions

DELFT3D

Deltares systems

DRAFT

WAVE

Deltares

Enabling Delta Life

User Manual

DRAFT

Delft3D-WAVE

Simulation of short-crested waves with SWAN

User Manual

Hydro-Morphodynamics

DRAFT

Version: 3.05
Revision: 41593

1 September 2015

Delft3D-WAVE, User Manual

DRAFT

Published and printed by:
Deltares
Boussinesqweg 1
2629 HV Delft
P.O. 177
2600 MH Delft
The Netherlands

telephone: +31 88 335 82 73
fax: +31 88 335 85 82
e-mail: info@deltares.nl
www: https://www.deltares.nl

For sales contact:
telephone: +31 88 335 81 88
fax: +31 88 335 81 11
e-mail: sales@deltaressystems.nl
www: http://www.deltaressystems.nl

For support contact:
telephone: +31 88 335 81 00
fax: +31 88 335 81 11
e-mail: support@deltaressystems.nl
www: http://www.deltaressystems.nl

Copyright © 2015 Deltares
All rights reserved. No part of this document may be reproduced in any form by print, photo print, photo copy, microfilm or any other means, without written permission from the publisher: Deltares.

Contents

1 A guide to this manual **1**
- 1.1 Introduction . . . 1
- 1.2 User manual . . . 1
- 1.3 Manual version and revisions . . . 2
- 1.4 Typographical conventions . . . 2
- 1.5 Changes with respect to previous versions . . . 3

2 Introduction to Delft3D-WAVE **5**
- 2.1 SWAN wave model . . . 5
  - 2.1.1 Introduction . . . 5
  - 2.1.2 Conceptual design of SWAN: an introduction . . . 5
  - 2.1.3 Coupling of SWAN with Delft3D . . . 6
- 2.2 Areas of application . . . 6
- 2.3 Standard features . . . 7
- 2.4 Special features . . . 7
- 2.5 Coupling to other modules . . . 7
- 2.6 Utilities . . . 7
- 2.7 Installation and computer configuration . . . 8

3 Getting started **9**
- 3.1 Overview of Delft3D-WAVE . . . 9
- 3.2 Main menu of Delft3D . . . 9
- 3.3 Getting into WAVE . . . 9
- 3.4 Exploring the menu options . . . 12
- 3.5 Exiting the WAVE-GUI . . . 13

4 Graphical User Interface **15**
- 4.1 Introduction . . . 15
- 4.2 MDW-file and attribute files . . . 15
- 4.3 Filenames and conventions . . . 16
- 4.4 Working with the WAVE-GUI . . . 17
- 4.5 Data groups of MDW-file . . . 18
  - 4.5.1 Description . . . 19
  - 4.5.2 Hydrodynamics . . . 19
  - 4.5.3 Grids . . . 20
    - 4.5.3.1 Computational grid . . . 21
    - 4.5.3.2 Bathymetry . . . 21
    - 4.5.3.3 Spectral resolution . . . 23
    - 4.5.3.4 Nesting . . . 25
    - 4.5.3.5 Hydrodynamics . . . 25
  - 4.5.4 Time frame . . . 26
  - 4.5.5 Boundaries . . . 29
  - 4.5.6 Obstacles . . . 36
  - 4.5.7 Physical parameters . . . 38
    - 4.5.7.1 Constants . . . 39
    - 4.5.7.2 Wind . . . 40
    - 4.5.7.3 Processes . . . 42
    - 4.5.7.4 Various . . . 44
  - 4.5.8 Numerical parameters . . . 45
  - 4.5.9 Output curves . . . 47
  - 4.5.10 Output parameters . . . 48
  - 4.5.11 Additional parameters . . . 52
- 4.6 Visualisation area window . . . 52

DRAFT

4.7 Help function 54

5 Running and post-processing **55**
5.1 Running 55
5.1.1 Standalone 55
5.1.2 Online with FLOW 55
5.1.3 Executing a scenario 56
5.1.4 Files and file sizes 57
5.1.5 Command-line arguments 58
5.2 Frequently asked questions 58
5.3 Post-processing 58
5.3.1 Introduction 58
5.3.2 Model result files of Delft3D-WAVE 59
5.3.3 Working with GPP 61
5.3.4 Working with Delft3D-QUICKPLOT 64

6 Tutorials **71**
6.1 Introduction 71
6.2 Siu-Lam wave model (1 grid; 3 wave runs) 71
6.2.1 Introduction 71
6.2.2 WAVE Graphical User Interface 72
6.2.3 Saving input data 73
6.2.4 Data groups 73
6.2.5 Description 73
6.2.6 Hydrodynamics 74
6.2.7 Grids 74
6.2.7.1 Computational grid 74
6.2.7.2 Bathymetry 75
6.2.7.3 Spectral resolution 75
6.2.7.4 Nesting 77
6.2.7.5 Hydrodynamics 77
6.2.8 Time frame 77
6.2.9 Boundaries 78
6.2.10 Obstacles 79
6.2.11 Physical parameters 80
6.2.11.1 Constants 81
6.2.11.2 Wind 82
6.2.11.3 Processes 82
6.2.11.4 Various 83
6.2.12 Numerical parameters 84
6.2.13 Output curves 84
6.2.14 Output parameters 85
6.2.15 Additional parameters 86
6.2.16 Executing the scenario 86
6.2.17 Output files of Delft3D-WAVE 87
6.2.18 Visualising results 87
6.3 Nested wave model 88
6.3.1 WAVE Graphical User Interface 88
6.3.1.1 Description 94
6.3.1.2 Hydrodynamics 94
6.3.1.3 Grids 94
6.3.1.4 Time frame 94
6.3.1.5 Boundaries 95
6.3.1.6 Obstacles 95

DRAFT

6.3.1.7 Physical parameters . . . 95
6.3.1.8 Numerical parameters . . . 95
6.3.1.9 Output curves . . . 95
6.3.1.10 Output parameters . . . 95
6.3.1.11 Additional parameters . . . 96
6.3.2 Run and postprocessing . . . 96
6.4 Online-WAVE coupling (including morphology) . . . 96
6.4.1 Introduction . . . 96
6.4.2 Delft3D-FLOW model . . . 98
6.4.2.1 Description . . . 98
6.4.2.2 Domain . . . 98
6.4.2.3 Time frame . . . 98
6.4.2.4 Processes . . . 98
6.4.2.5 Initial conditions . . . 99
6.4.2.6 Boundaries . . . 99
6.4.2.7 Physical parameters . . . 100
6.4.2.8 Numerical parameters . . . 100
6.4.2.9 Operations . . . 100
6.4.2.10 Monitoring . . . 100
6.4.2.11 Additional parameters . . . 100
6.4.2.12 Output . . . 101
6.4.3 Delft3D-WAVE model . . . 101
6.4.3.1 Description . . . 101
6.4.3.2 Hydrodynamics . . . 102
6.4.3.3 Grids . . . 102
6.4.3.4 Time frame . . . 102
6.4.3.5 Boundaries . . . 102
6.4.3.6 Obstacles . . . 103
6.4.3.7 Physical parameters . . . 103
6.4.3.8 Numerical parameters . . . 103
6.4.3.9 Output curves . . . 103
6.4.3.10 Output parameters . . . 103
6.4.4 Run and postprocessing . . . 104
6.4.4.1 Foreground . . . 104
6.4.4.2 Background . . . 104
6.4.4.3 Output files . . . 105
6.5 FLOW-DD and Online WAVE . . . 105
6.5.1 Introduction . . . 105
6.5.2 Delft3D-FLOW models . . . 106
6.5.2.1 Model set-up outside FLOW domain . . . 106
6.5.2.2 Description . . . 106
6.5.2.3 Domain . . . 107
6.5.2.4 Time frame . . . 107
6.5.2.5 Processes . . . 107
6.5.2.6 Initial conditions . . . 107
6.5.2.7 Boundaries . . . 107
6.5.2.8 Physical parameters . . . 108
6.5.2.9 Numerical parameters . . . 108
6.5.2.10 Operations . . . 108
6.5.2.11 Monitoring . . . 108
6.5.2.12 Additional parameters . . . 108
6.5.2.13 Output . . . 108
6.5.2.14 Model set-up inside FLOW domain . . . 108
6.5.2.15 Description . . . 109

DRAFT

6.5.2.16 Domain . . . 109
6.5.2.17 Boundaries . . . 110
6.5.3 Delft3D-WAVE model . . . 110
6.5.3.1 Description . . . 110
6.5.3.2 Hydrodynamics . . . 111
6.5.3.3 Grids . . . 111
6.5.3.4 Time frame . . . 111
6.5.3.5 Boundaries . . . 111
6.5.3.6 Obstacles . . . 112
6.5.3.7 Physical parameters . . . 112
6.5.3.8 Numerical parameters . . . 112
6.5.3.9 Output curves . . . 112
6.5.3.10 Output parameters . . . 113
6.5.4 Run and postprocessing . . . 113
6.5.4.1 Foreground . . . 113
6.5.4.2 Background . . . 113
6.5.4.3 Output files . . . 115

7 Conceptual description 117
7.1 Introduction . . . 117
7.2 General background . . . 117
7.2.1 Units and co-ordinate systems . . . 117
7.2.2 Choice of grids and boundary conditions . . . 118
7.2.3 Output grids . . . 120
7.3 Physical background of SWAN . . . 120
7.3.1 Action balance equation . . . 120
7.3.2 Propagation through obstacles . . . 124
7.3.3 Wave-induced set-up . . . 125
7.3.4 Diffraction . . . 125
7.4 Full expressions for source terms . . . 125
7.4.1 Input by wind . . . 125
7.4.2 Dissipation of wave energy . . . 126
7.4.3 Nonlinear wave-wave interactions . . . 128
7.5 Numerical implementation . . . 131
7.5.1 Propagation . . . 132

References 135

A Files of Delft3D-WAVE 141
A.1 MDW-file . . . 141
A.1.1 General description . . . 141
A.1.2 Offline calculation . . . 145
A.2 Attribute files of Delft3D-WAVE . . . 145
A.2.1 Introduction . . . 145
A.2.2 Orthogonal curvilinear grid . . . 146
A.2.3 Time-series for wave boundary conditions . . . 147
A.2.4 Obstacle file . . . 147
A.2.5 Segment file . . . 149
A.2.6 Depth file . . . 150
A.2.7 Space-varying bottom friction (not yet implemented for Delft3D-WAVE) 151
A.2.8 Wave boundary conditions . . . 152
A.2.8.1 Time-varying and uniform wave conditions in <wavecon.*rid*> file . . . 152

DRAFT

A.2.8.2 Time-varying and space-varying wave boundary conditions using BCW files . . . 153
A.2.8.3 Space-varying wave boudnary conditions using for UNIBEST coupling (<md-vwac>-file) . . . 160
A.2.8.4 Time- and space-varying wave boundary conditions: TPAR file . . . 161
A.2.9 Spectral input and output files . . . 161
A.2.10 Space-varying wind field . . . 166
A.2.10.1 Space-varying wind on the computational (SWAN) grid . . . 168
A.2.10.2 Space-varying wind on an equistant grid . . . 171
A.2.10.3 Space-varying wind on a curvilinear grid . . . 175
A.2.10.4 Space-varying wind on a Spiderweb grid . . . 177

B Definition of SWAN wave variables **183**

C Example of MDW-file Siu-Lam **187**

D DATSEL data extraction utility **189**
D.1 Function . . . 189
D.2 Running DATSEL . . . 189
D.3 Input description . . . 189
D.4 Output files . . . 191
D.5 Example file . . . 192

E LINT Line Integration **193**
E.1 Function . . . 193
E.2 Running LINT . . . 193
E.3 Input description . . . 193
E.4 Output files . . . 194
E.5 Example file . . . 194

F KUBINT volume integration **197**
F.1 Function . . . 197
F.2 Running KUBINT . . . 197
F.3 Input description . . . 197
F.4 Output files . . . 198
F.5 Example file . . . 198

DRAFT

DRAFT

List of Figures

3.1 Main window Delft3D-MENU . . . 9
3.2 Selection window for Waves . . . 10
3.3 **Select working directory** window . . . 10
3.4 Main window of the WAVE Graphical User Interface . . . 11
3.5 Menu bar options in the WAVE-GUI . . . 12
3.6 *File* menu options . . . 12
3.7 *View* menu option . . . 12
3.8 *Help* menu option . . . 12
3.9 Canvas with input fields and selection buttons for the Data Group *Boundaries* . 13

4.1 Options in the main window of the WAVE Graphical User Interface . . . 17
4.2 Window of Data Group *Description* . . . 19
4.3 Data Group *Hydrodynamics* . . . 20
4.4 Data Group *Grids*, sub-group *Computational grid* . . . 21
4.5 Data Group *Grids*, sub-group *Bathymetry* . . . 22
4.6 Data Group *Grids*, sub-group *Spectral resolution* . . . 24
4.7 Data Group *Grids*, sub-group *Nesting* . . . 25
4.8 Data Group *Grids*, sub-group *Hydrodynamics* . . . 26
4.9 Data Group *Grids*, sub-group *Hydrodynamics* . . . 27
4.10 Data Group *Time frame* in case of standalone WAVE computation . . . 28
4.11 Data Group *Time frame*, using FLOW results . . . 29
4.12 Data Group *Boundaries* . . . 30
4.13 Boundary orientations . . . 31
4.14 Definition of boundary using XY coordinates . . . 31
4.15 Window **Uniform boundary conditions**. After pressing *Edit Conditions* when *Uniform* and *Parametric* where selected . . . 33
4.16 Window **Space-varying boundary conditions**. After pressing *Edit Conditions* when *Space-varying* and *Parametric* where selected . . . 33
4.17 Window **Space-varying boundary conditions**. After pressing *Edit Conditions* when *Space-varying* and *Parametric* where selected. . . . 34
4.18 Window **Space-varying boundary conditions**. After pressing *Edit spectral space* when *Space-varying* and *Parametric* where selected. . . . 35
4.19 Data Group *Obstacles* . . . 36
4.20 Data Group *Physical parameters* . . . 38
4.21 Sub-data Group *Physical parameters, Constants* . . . 39
4.22 Sub-data Group *Physical parameters, Wind* . . . 41
4.23 Sub-data Group *Physical parameters, Processes* . . . 42
4.24 Sub-data Group *Physical parameters, Various* . . . 45
4.25 Data Group *Numerical parameters* . . . 46
4.26 Data Group *Output curves* . . . 48
4.27 Data Group *Output parameters* . . . 49
4.28 Data Group *Output parameters: Output locations* . . . 51
4.29 Data Group *Additional parameters* . . . 52
4.30 Canvas with Visualisation Area of the wave module . . . 53
4.31 *File - Open* menu options . . . 53
4.32 *File - Print area* menu options . . . 53

5.1 **Waves (standalone)** selection window for executing a scenario . . . 55
5.2 **Hydrodynamics** selection window to execute a FLOW-WAVE simulation . . . 56
5.3 Select scenario to be executed . . . 56
5.4 Hierarchy of GPP . . . 61
5.5 Main window of GPP . . . 62

DRAFT

5.6 Parameters and locations in the <trih-tut_fti.dat> file . . . 64
5.7 Plot window of GPP . . . 65
5.8 Delft3D-QUICKPLOT main window . . . 66
5.10 User interface after opening a Delft3D-WAVE map file . . . 67
5.11 List of data fields in the Delft3D-WAVE map file . . . 67
5.12 List of plot options is changed after selection of the *hsig wave height* from the dropdown list . . . 68
5.13 Optional listing of the times associated with the various time steps . . . 68
5.14 Selection of a cross-section along a grid line in M direction: one M value, all N values . . . 69
5.15 2D Plot of the 'hsig wave height . . . 70

6.1 Starting window of the WAVE Graphical User Interface . . . 72
6.2 Data Group: *Description* and sub-window . . . 73
6.3 Data Group *Grids* . . . 74
6.4 **Visualisation Area** window . . . 75
6.5 Sub-data Group *Bathymetry* . . . 76
6.6 Sub-data Group *Spectral resolution* . . . 76
6.7 Data Group *Time frame*. . . . 77
6.8 Data Group: *Boundaries* . . . 78
6.9 Space-varying boundary conditions . . . 79
6.10 Spectral space input parameters . . . 80
6.11 Data Group *Obstacles* . . . 81
6.12 Sub-data Group *Constants* . . . 81
6.13 Sub-data Group: *Wind* . . . 82
6.14 Sub-data Group *Processes* . . . 82
6.15 Sub-data Group *Various* . . . 83
6.16 Data Group *Numerical parameters* . . . 84
6.17 Data Group *Output parameters* . . . 85
6.18 **Output locations** window . . . 86
6.19 Select scenario to run . . . 87
6.20 Top panel: Siu Lam model area near Hong Kong area. Bottom panel: LAND BOUNDARY and curvilinear flow GRID . . . 89
6.21 Top panel: Model BATHYMETRY of Siu Lam model. Bottom panel: BATHYMETRY and GRID of Siu Lam model . . . 90
6.22 Top panel: Computed WAVE HEIGHT pattern on 1 Oct 2005, 18:00. Bottom panel: Computed MEAN WAVE PERIOD pattern on 1 Oct 2005, 18:00 . . . 91
6.23 Top panel: Computed ENERGY TRANSPORT on 1 Oct 2005, 18:00. Bottom panel: Computed DISSIPATION pattern on 1 Oct 2005, 18:00 . . . 92
6.24 Top panel: WAVE vector on 1 Oct 2005, 18:00. Bottom panel: Significant WAVE HEIGHT on 1 Oct 2005, 18:00 . . . 93
6.25 Data Group *Grids – Nesting* window . . . 95
6.26 Data group *Output parameters*, output for computational grids . . . 96
6.27 Development of a spit at the Head of Ameland: the Bornrif . . . 96
6.28 Measured 1989 and 1996 bathymetry . . . 97
6.29 Output restrictions for Online-WAVE . . . 98
6.30 Overview of active processes . . . 99
6.31 Overview of output parameters . . . 101
6.32 Overview of output parameters in Delft3D-WAVE . . . 103
6.33 Execute the Flow-Wave model . . . 104
6.35 Wind drag coefficients in Delft3D-FLOW for outside domain set-up . . . 108
6.36 Numerical parameters in Delft3D-FLOW for outside domain setup . . . 109
6.37 Overview of output parameters of the Delft3D-FLOW model for the outside domain . . . 110

DRAFT

6.38 Wave boundary conditions for 'Boundary North' in the WAVE model set-up . . 112
6.39 Numerical parameters used in the WAVE model set-up 114
6.40 Overview of out parameters in Delft3D WAVE. 114

7.1 Nautical convention (left panel) and Cartesian convention (right panel) for direction of winds and (incident) waves 117
7.2 Definition of grids (input, computational and output grids) in Delft3D-WAVE . . 118
7.3 Disturbed regions in the computational grid 119

A.1 Definition wind components for space varying wind 171
A.2 Definition sketch of wind direction according to Nautical convention 173
A.3 Illustration of the data to grid conversion for meteo input on a separate curvilinear grid 178
A.4 Wind definition according to Nautical convention 180
A.5 Spiderweb grid definition 180

D.1 Selection window for Waves Tools 189

E.1 Selection window for Waves Tools 193

F.1 Selection window for Morphology Tools 197

DRAFT

DRAFT

List of Tables

5.1 Output parameters in <wavm-∗.dat> 59
5.2 Output parameters in <com-∗.dat> 60

DRAFT

DRAFT

1 A guide to this manual

1.1 Introduction

To simulate the evolution of wind-generated waves in coastal waters (which may include estuaries, tidal inlets, barrier islands with tidal flats, channels etc.) the Delft3D-WAVE module can be used. The wave module of Delft3D computes wave propagation, wave generation by wind, non-linear wave-wave interactions and dissipation, for a given bottom topography, wind field, water level and current field in waters of deep, intermediate and finite depth.

At present two wave models (both of the phase-averaged type) are available in Delft3D. They are the second-generation HISWA wave model (Holthuijsen *et al.*, 1989) and, its successor, the third-generation SWAN wave model (Booij *et al.*, 1999; Ris *et al.*, 1999).

The SWAN wave model is presently the **standard option** within Delft3D.

1.2 User manual

In this manual advice is given on how to get started with the SWAN wave model of Delft3D. Furthermore, the manual gives a description on how to use the SWAN model within Delft3D-WAVE.

Generally, the following items with respect to the use of the Delft3D-WAVE module will be described in this manual:

Chapter 2: Introduction to Delft3D-WAVE, provides specifications of Delft3D-WAVE such as required computer configuration, how to install the software, as well as its main features.

Chapter 3: Getting started, explains the use of the overall menu program, which gives access to all Delft3D modules and to the pre-processing and post-processing tools. A first introduction is given into the WAVE Graphical User Interface (GUI), used to define the input required for a wave simulation.

Chapter 4: Graphical User Interface, provides practical information on the selection of all parameters and the tuning of the model.

Chapter 5: Running and post-processing, discusses how to execute a scenario and visualise the results. Information on run times and file sizes is given as well as a brief introduction to the post-processing programs GPP and Delft3D-QUICKPLOT, which can be used to visualise the simulation results of the wave module.

Chapter 6: Tutorials, emphasises at giving you some first hands-on experience in using the WAVE Graphical User Interface to define the input of a simple problem, in verifying this input, in executing the simulation and in inspecting the results.

Chapter 7: Conceptual description, discusses the unit and co-ordinate system, the various grids, grid-numbering etc. In addition, a brief description is given on the physics and numerics that have been implemented in the wave module of Delft3D.

References, provides a list of publications and related material on the Delft3D-WAVE module.

Appendix A: Files of Delft3D-WAVE, gives a description of all the attribute files that can be used in the Delft3D-WAVE input. This information is required for generating certain attribute files either manually or by means of other utility programs. For other attribute files this

DRAFT

description is just for your information.

Appendix B: Definition of SWAN wave variables, the definition of the integral wave parameters is given.

Appendix C: Example of MDW-file Siu-Lam, an example of a Master Definition file for the Wave <∗.mdw> input file for the WAVE module is given.

Appendix D: DATSEL data extraction utility contains the User Manual for the data extraction utility DATSEL.

Appendix E: LINT Line Integration contains the User Manual for the line integration program LINT.

Appendix F: KUBINT volume integration contains the User Manual for the kubing program KUBINT.

1.3 Manual version and revisions

The version number and the release date of this User Manual are given in the top right corner on each page. Revisions to this manual will be indicated by the version number followed by the revision number separated by a dot, for example version 3.00. A revision number of this manual will not necessarily be the same as the revision number of the module it concerns. This manual describes the functionality of WAVE 1.04.09 and WAVE-GUI version 4.92.00.

1.4 Typographical conventions

Throughout this manual, the following conventions in text formats help you to distinguish between different types of text elements.

Example	Description
Waves **Boundaries**	Title of a window or sub-window. Sub-windows are displayed in the **Module** window and cannot be moved. Windows can be moved independently from the **Module** window, such as the **Visualisation Area** window.
Save	Item from a menu, title of a push button or the name of a user interface input field. Upon selecting this item (click or in some cases double click with the left mouse button on it) a related action will be executed; in most cases it will result in displaying some other (sub-)window. In case of an input field you are supposed to enter input data of the required format and in the required domain.
<\tutorial\wave\swan-curvi> <siu.mdw>	Directory names, filenames, and path names are expressed between angle brackets, <>. For the Linux and UNIX environment a forward slash (/) is used instead of the backward slash (\) for PCs.

DRAFT

Example	Description
“27 08 1999”	Data to be typed by you into the input fields are displayed between double quotes. Selections of menu items, option boxes etc. are described as such: for instance ‘select *Save* and go to the next window’.
`delft3d-menu`	Commands to be typed by you are given in the font Courier New, 10 points.
	User actions are indicated with this arrow.
[m/s] [-]	Units are given between square brackets when used next to the formulae. Leaving them out might result in misinterpretation.

1.5 Changes with respect to previous versions

Version	Description
3.05	Description of <bcw>-file added (section A.2.8), this file type can be used with Delft3D-WAVE version 3.04.01.1869, which version can be downloaded from https://svn.oss.deltares.nl/repos/delft3d/trunk/src
3.04	Some elaborations on diffraction and non-stationary computations. Additional output on wavm file
3.03	Chapter 6, Figure 6.23 and Figure 6.24 changed (now parameters from wavm-file). Chapter 6 Tutorials. Tutorial 4 is added. This tutorial concerns the coupling of Delft3D-FLOW Domain Decomposition and Delft3D-WAVE. Chapter 6, Tutorials; For all boundary conditions the directional spreading is converted to 4 [-] (meaning cosine power), instead of 4 [degrees]. Appendices E and F: The polylines for LINT and KUBINT are not anymore specified explicitly, but by specifying the filename which contains the polylines. The old input is not supported anymore from v2.00.00 and higher.

DRAFT

Version	Description
3.02	In Section 5.1 how to run standalone and online with FLOW described. Section 5.1.5 Command-line arguments added. The WAVE-GUI is improved concerning its layout. New functionality non-stationary wave simulations. New functionality wind parameters from FLOW simulation. New functionality extending the (individual) parameters from the com-file to cover the whole WAVE grid. New functionality when running WAVE online with FLOWthe interval for writing to the wavm-file can be specified. Default value for 'Directional spreading' changed from '0' to '4'. In DG *Description* WAVE and MOR file lines removed. The opened mdw-filename is shown in the titlebar. If the wind speed is larger than zero, and in Sub-data Group *Processes* the third generation mode is selected, then the Quadruplets in Sub-data Group *Various* will be activated. Various MENU screens updated.
3.01	New functionality reflection at obstacles described. New functionality diffraction described. New functionality hotfile described. Tutorial for combined hydrodynamics, morphology and waves added. Online coupling with FLOW described. To start a wave simulation, an $<*$.mdw$>$ file has to be selected. The $<*$.mdm$>$ file is obsolete. Convert old files first with WAVE-GUI 4.90.00 or higher.
3.00	New input files: $<$md-wave.$*>$ changed to $<*$.mdw$>$ and $<$morf.$*>$ changed to $<*$.mdm$>$ Data Group *Grids* redesigned in GUI; curvilinear nested grids supported. Data Group *Tidal information* removed from GUI. Data Group *Obstacles* extended with *Import from file*. **Visualisation Area** window updated. Double precision RGFGRID grids accepted. Description of HISWA removed from the User Manual. User Manual of DATSEL, LINT and KUBINT added as appendices. Tutorials revised.
2.10	Improved layout of manual.

DRAFT

2 Introduction to Delft3D-WAVE

2.1 SWAN wave model

2.1.1 Introduction

To simulate the evolution of random, short-crested wind-generated waves in estuaries, tidal inlets, lakes etc., the ***third-generation*** SWAN model - SWAN is an acronym for **S**imulating **WA**ves **N**earshore - can be used (see e.g. Holthuijsen *et al.* (1993); Booij *et al.* (1999); Ris *et al.* (1999)).

This SWAN model is the successor of the stationary second-generation HISWA model. The SWAN model has a number of advantages compared to HISWA and also overcomes to a large extent the limitations of the HISWA model. The main characteristics of SWAN with respect to the physics and numerics are:

1 The physics in SWAN are explicitly represented with state-of-the-art formulations
2 The SWAN model is fully spectral in frequencies and directions (0°–360°)
3 The wave computations in SWAN are unconditionally stable due to the fully implicit schemes that have been implemented
4 The computational grid in SWAN has *not to be oriented* in the mean wave direction and so the grid can handle all wave directions.

Other aspects, which may be of importance in practical applications of the Delft3D-WAVE module, are:

1 SWAN can perform computations on a curvilinear grid (if the FLOW module of Delft3D uses this grid, the coupling between SWAN and FLOW is perfect).
2 The wave forces can also be computed on the gradient of the radiation stress tensor (rather than on the dissipation rate as in the HISWA model).
3 Output can be generated in terms of one- and two-dimensional wave spectra in SWAN.

2.1.2 Conceptual design of SWAN: an introduction

The SWAN model is based on the discrete spectral action balance equation and is fully spectral (in all directions and frequencies). The latter implies that short-crested random wave fields propagating simultaneously from widely different directions can be accommodated (e.g. a wind sea with super-imposed swell). SWAN computes the evolution of random, short-crested waves in coastal regions with deep, intermediate and shallow water and ambient currents. The SWAN model accounts for (refractive) propagation due to current and depth and represents the processes of wave generation by wind, dissipation due to whitecapping, bottom friction and depth-induced wave breaking and non-linear wave-wave interactions (both quadruplets and triads) explicitly with state-of-the-art formulations. Wave blocking by currents is also explicitly represented in the model.

To avoid excessive computing time and to achieve a robust model in practical applications, fully implicit propagation schemes have been applied. The SWAN model has successfully been validated and verified in several laboratory and (complex) field cases (see Ris *et al.* (1999); WL | Delft Hydraulics (1999, 2000)).

The SWAN model was developed at Delft University of Technology (The Netherlands). It is specified as the new standard for nearshore wave modelling and coastal protection studies. It is therefore that Deltares is integrating the SWAN model in the Delft3D model suite. The SWAN model has been released under public domain. For more information about SWAN

DRAFT

reference is made to the SWAN home page:

http://fluidmechanics.tudelft.nl/swan/default.htm

2.1.3 Coupling of SWAN with Delft3D

When it comes to taking into account the effect of flow on the waves (via set-up, current refraction and enhanced bottom friction) and the effect of waves on current (via forcing, enhanced turbulence and enhanced bed shear stress), there are three different types of wave computations within the Delft3D module:

1. a WAVE computation that uses user-defined flow properties: for each wave condition, you specify a spatially uniform water level and a spatially uniform current velocity, so that the effect of flow on waves is accounted for,
2. an offline coupling of WAVE with Delft3D-FLOW: the wave computation uses flow characteristics from a completed Delft3D-FLOW computation, so that the effect of flow on waves is accounted for,
3. an online coupling of WAVE with Delft3D-FLOW: the WAVE model has a dynamic interaction with the FLOW module of Delft3D (i.e. two way wave-current interaction). Through this coupling, both the effect of waves on current and the effect of flow on waves are accounted for.

Besides the three types of wave computation mentioned above, it is also possible to run a WAVE computation, where the influence of flow characteristics on the waves in the model area is not accounted for.

In case the offline coupling (type 2) or the online coupling (type 3, dynamic interaction) between the FLOW and WAVE module of Delft3D is used, data is exchanged using a so-called communication file (com-file), which contains the most recent data of the flow and wave computations.

2.2 Areas of application

The SWAN model of Delft3D-WAVE can be used for coastal development and management related projects and for harbour and offshore installation design. It can also be used as a wave hindcast model. Typical areas for the application of the SWAN model may vary of up to more than 50 km $\times$ 50 km. Generally, the model can be applied in the following areas:

- ◇ estuaries
- ◇ tidal inlets
- ◇ lakes
- ◇ barrier islands with tidal flats
- ◇ channels
- ◇ coastal regions

DRAFT

2.3 Standard features

The SWAN model accounts for the following physics:

- ◇ wave refraction over a bottom of variable depth and/or a spatially varying ambient current
- ◇ depth and current-induced shoaling
- ◇ wave generation by wind
- ◇ dissipation by whitecapping
- ◇ dissipation by depth-induced breaking
- ◇ dissipation due to bottom friction (three different formulations)
- ◇ nonlinear wave-wave interactions (both quadruplets and triads)
- ◇ wave blocking by flow
- ◇ transmission through, blockage by or reflection against obstacles
- ◇ diffraction

Note that diffraction and reflections are now available in the present SWAN version under Delft3D-WAVE.

2.4 Special features

A special feature is the dynamic interaction with the FLOW module of Delft3D (i.e. two way wave-current interaction). By this the effect of waves on current (via forcing, enhanced turbulence and enhanced bed shear stress) and the effect of flow on waves (via set-up, current refraction and enhanced bottom friction) are accounted for.

2.5 Coupling to other modules

The wave conditions (i.e. wave forces based on the energy dissipation rate or the radiation stresses, orbital bottom velocity) calculated in the Delft3D-WAVE module are used as input for the other modules of Delft3D, which are:

module	description
Delft3D-FLOW	wave driven currents, enhanced turbulence and bed shear stress.
Delft3D-FLOW 3DMOR	stirring by wave breaking.

2.6 Utilities

For using Delft3D-WAVE, the following utilities are important:

module	description
RGFGRID	for generating grids.
QUICKIN	for preparing and manipulating grid oriented data, such as bathymetry or initial conditions for water levels.
GPP	for visualising simulation results.
Delft3D-QUICKPLOT	for visualising simulation results.

For details on using these utility programs you are referred to the respective User Manual (RGFGRID, 2013; QUICKIN, 2013; GPP, 2013; QUICKPLOT, 2013).

DRAFT

2.7 Installation and computer configuration

See the Delft3D Installation Manual (Delft3D-IM, 2013).

DRAFT

3 Getting started

3.1 Overview of Delft3D-WAVE

Delft3D is a range of modules which can be run independently of one another. Therefore, the modules are supplied separately. The modules are provided with a menu shell through which you can access the various modules, WAVE being one of them. We will now guide you through some of its screens to get the look-and-feel of the program. For a more detailed description of the program, you are referred to Chapter 4. Later on, in Chapter 6, you can run a simple scenario by following the instructions in a tutorial.

3.2 Main menu of Delft3D

The main menu of Delft3D gives access to all modules of Delft3D, including Delft3D-WAVE. To arrive at this menu you should:

- ◇ In Windows XP or Windows NT select Delft3D in the *Applications* Menu or click on the Delft3D-MENU icon on the desk-top.
- ◇ On Linux machines type: `delft3d-menu` on the command line.

Next the window containing the Delft3D-MENU appears (see Figure 3.1).

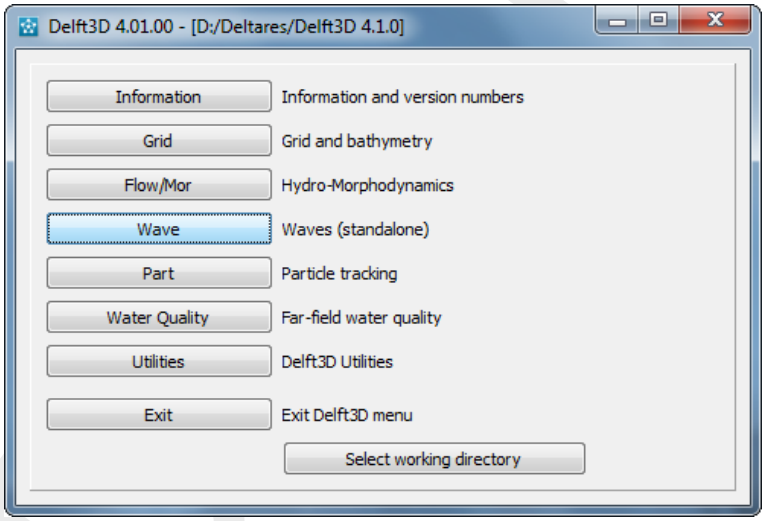

Figure 3.1: *Main window Delft3D-MENU*

Remark:

- ◇ In this and the following chapters several windows are shown to illustrate the presentation of Delft3D-MENU and Delft3D-WAVE. These windows are grabbed from the PC-platform. For Linux the content of the windows is the same, but the colours may be different.

3.3 Getting into WAVE

To select the Delft3D-WAVE module just:

- ◇ Click the *WAVE* button

Next the selection window pops up for preparing a wave input file <∗.mdw>, to execute a computation in the foreground or background, to inspect the monitoring files with information

on the execution and to visualise the results (see Figure 3.2).

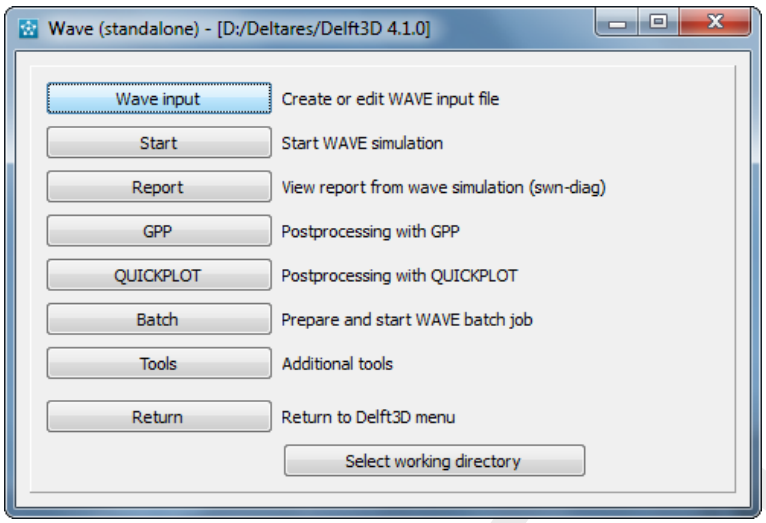

Figure 3.2: *Selection window for Waves*

Before continuing with any of the selections of this **Waves (standalone)** window, you must select the directory in which you are going to prepare scenarios and execute computations:

◇ Click the *Select working directory* button, see Figure 3.3 for the window displayed.

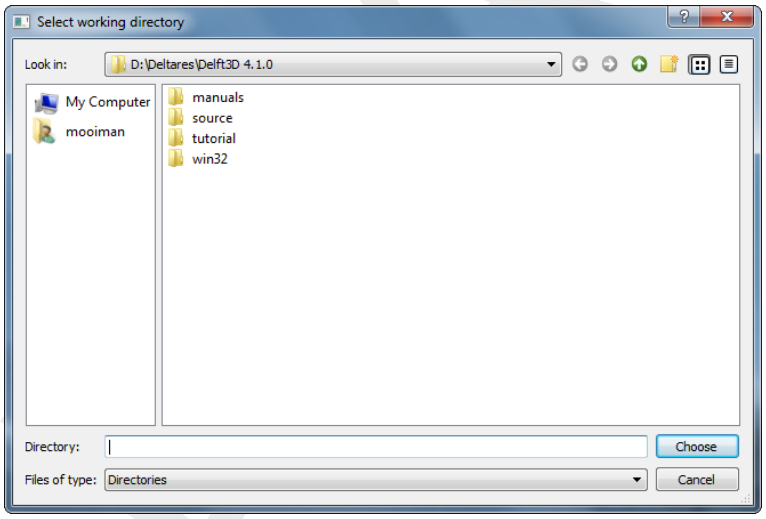

Figure 3.3: Select working directory *window*

A standard file selection window is opened and you can navigate to the required directory.

◇ Browse to the desired directory, and enter this working directory.
◇ Confirm your selection by clicking *OK*.

Remark:

◇ In case you want to create a new directory, click and specify a name. Enter the new directory and click *OK* to confirm your selection.

Now we are back in the main wave menu and we can define and execute a scenario. In this

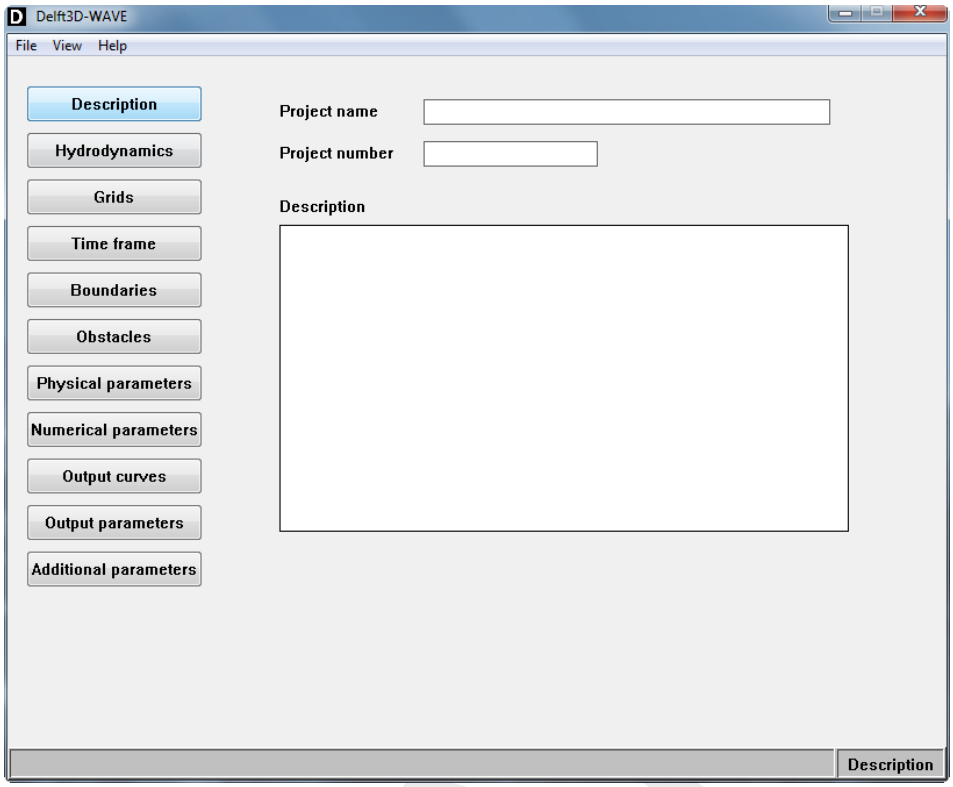

Figure 3.4: *Main window of the WAVE Graphical User Interface*

guided tour through Delft3D-WAVE we limit ourselves to “create or edit a WAVE input file”, since this is the main user task for the WAVE Graphical User Interface (GUI), hence:

◇ Click on *Wave input.*

The WAVE-GUI is loaded and the primary input screen is opened, see Figure 3.4.

DRAFT

3.4 Exploring the menu options

The items at the far most left of the menu bar can be handled as any other item in a Window-oriented menu. After starting the WAVE-GUI you have the following options, see Figure 3.5:

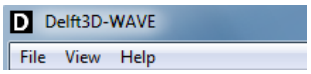

Figure 3.5: *Menu bar options in the WAVE-GUI*

File	select and open an mdw-file, save an mdw-file, save an mdw-file under a different name or 'exit' the WAVE-GUI.
View	visualisation area.
Help	About information

◇ Clicking on *File* enables several options, see Figure 3.6:

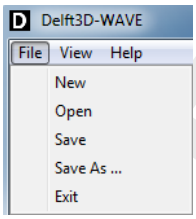

Figure 3.6: File *menu options*

The *File* menu in the main window allows you to read or write an input file.

New	create a new input file
Open	to open an existing input file with the purpose to inspect or change it. The input file contains the wave information.
Save	to save input files under the same name after it has been modified.
Save As	to save input files under a different name.
Exit	to exit the WAVE-GUI and return to the **Waves** Selection window program. Save your results first! No warning will be given.

◇ Clicking on *View* enables one option, see Figure 3.7:

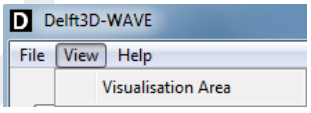

Figure 3.7: View *menu option*

The View menu in the main window allows you to open a Visualisation Area window.

◇ Clicking on *Help* enables only the *About* option (see Figure 3.8), which provides information on the version of the User Interface.

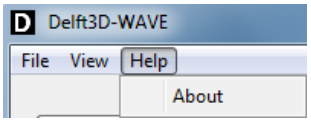

Figure 3.8: Help *menu option*

DRAFT

The input parameters that define a Delft3D-WAVE model are grouped into data groups. These groups are represented by the large grey buttons at the left of the main window. Clicking on a data group will result in a canvas area where the data can be filled in. This canvas area will be dynamically filled with input fields, tables, or list boxes to define the various kinds of input data required for a simulation. Click on them to see what happens next. For example, clicking the *Boundaries* button, and next press the *Add* button, will result in the window shown in Figure 3.9.

The Tutorials in Chapter 6 will make you become fully acquainted with the various input windows that result from this main window.

You are encouraged to explore the various data groups and sub-windows to get a first impression of the items the data groups are composed of. Though several input items are related there is no fixed or prescribed order in defining the input data. Occasionally you will get a warning or error message that some data is not saved or not consistent with earlier defined data; just neglect these messages and press the *OK* button if requested. No harm will be done on existing input files as you are not going to save the input data of this exercise.

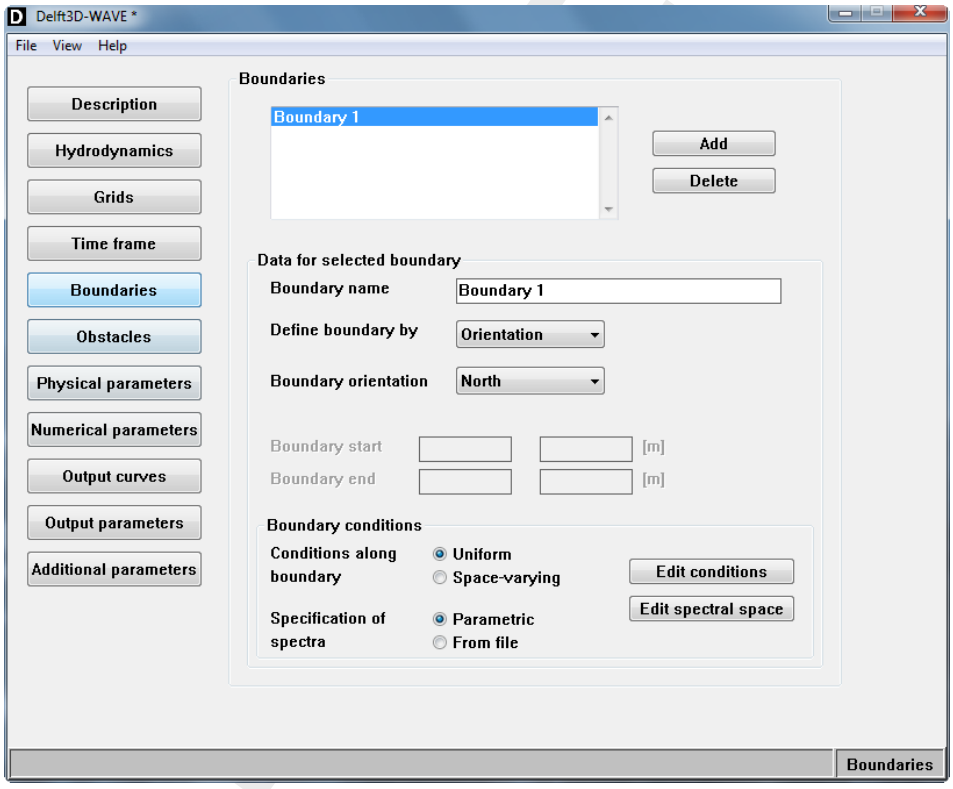

Figure 3.9: *Canvas with input fields and selection buttons for the Data Group* Boundaries

3.5 Exiting the WAVE-GUI

To exit the WAVE-GUI:

◇ Click *File* → *Exit.*

You will be back in the **Waves** selection window, see Figure 3.2.

Now, ignore the other options and just:

◇ Click *Return* to return to the main window of Delft3D-MENU, see Figure 3.1.

◇ Click *Exit.*

The window is closed and the control is returned to the desktop or the command line.

This Getting Started session will have given you the general idea of how to access the WAVE-GUI and how to load an existing input mdw-file.

DRAFT

4 Graphical User Interface

4.1 Introduction

In order to set up a wave model you must prepare an input file. The input file stores all the parameters used for a wave computation with Delft3D-WAVE. The parameters can be divided into three categories:

1 parameters that define the physical processes being modelled,
2 parameters that define the numerical techniques used to solve the equations that describe the physical processes,
3 parameters that control the wave computation and store its results.

Within the range of realistic values, it is likely that the solution is sensitive to the selected parameter values, so a concise description of all parameters is required. The input data (defined by you) is stored into an input file which is called the Master Definition file for Wave or MDW-file.

In section 4.2 we discuss some general aspects of the MDW-file and its attribute files. section 4.3 discusses shortly the filenames and their extension. In section 4.4 we explain how to work with the WAVE Graphical User Interface. In section 4.5 all input parameters are discussed, including their restrictions and their valid ranges or domain. Finally, in Sections 4.6 and 4.7, it is explained how to deal with the so-called 'Visualisation Area Window' and the help function, respectively.

4.2 MDW-file and attribute files

The Master Definition Wave file (MDW-file) is the input file for the wave program. It contains all the necessary data that is required to define a wave model and run a wave computation. Some of the parameter values are given directly in the MDW-file. Other parameters are defined in attribute files, referred to by specific statements in de MDW-file. The latter is particularly the case when parameters contain a large number of data (e.g. spatially varying data such as a variable wind or friction field). The user-defined attribute files are listed and described in Appendix A.

The WAVE Graphical User Interface, or WAVE-GUI (see Figure 3.4), is a tool that is used to assign values to all the necessary parameters or to import the names of the attribute files into the MDW-file. When the data you entered is saved (see Figure 3.6), an mdw-file, containing all the specified data, is created in the selected working directory.

Although you are not supposed to work directly on the mdw-file (with a text editor) it is useful to have some idea of what its structure is, as it reflects the idea of the designer on how to handle large amounts of input data. For an example of an MDW-file, see Appendix C.

The basic characteristics of an MDW-file are:

- It is an ASCII file.
- The file is divided in datagroups.
- It is keyword based.

The mdw-file is an intermediate file between the WAVE Graphical User Interface and the Delft3D-WAVE module. As it is an ASCII-file, it can be transported to an arbitrary hardware platform. Consequently, the wave module and the WAVE Graphical User Interface program

DRAFT

do not necessarily have to reside in the same hardware platform.

As explained before (and you will also see this in Chapter 6), input parameters that contain a lot of data are defined in attribute files. You have to set up these attribute files outside the WAVE-GUI, before they can be imported into the mdw-file. How to set up these attribute files is explained elsewhere in this chapter. The mdw-file only contains permanent input parameters and references to these attribute files. The formats of all attribute files (and of the mdw-file itself) are described in detail in Appendix A.

The mdw-file and its attribute files form a complete set, defining a simulation. When storing your simulation input, always make sure you include the complete set of MDW-file and attribute files.

4.3 Filenames and conventions

The names of the mdw-file and its attribute files have a specific structure, some aspects are obliged while others are only advised or preferred.

The name of an mdw-file must have the following structure: <run-id.mdw>. The <run-id> consists of an arbitrary combination of (maximum 252) letters and numbers. This <run-id> will be part of the result files to safeguard the link between an mdw-file and the result files.

Restriction:

- ◇ The maximum length of the <run-id> is 252 characters!

The names of the attribute files follow the general file naming conventions, i.e. they have the following structures: <name>.<extension>. Where:

- <name> is any combination of characters allowed for filenames, except spaces.
- There is no limitation other than the platform dependent limitations; you are referred to your platform manual for details. We suggest to add some continuation character, for instance <-number> to the <name> to distinguish between various updates or modifications of the file.
- The <extension> is mandatory as indicated below.

Quantity	Filename and mandatory extension
Bathymetry or water depth	<name>.dep
Curvilinear grid	<name>.grd
Grid enclosure	<name>.enc
Wind field	<name>.wnd
Spectral wave boundary	<name>.bnd
Curves	<name>.pol
Output locations	<name>.loc
Obstacles	<name>.obs
Obstacles locations	<name>.pol

DRAFT

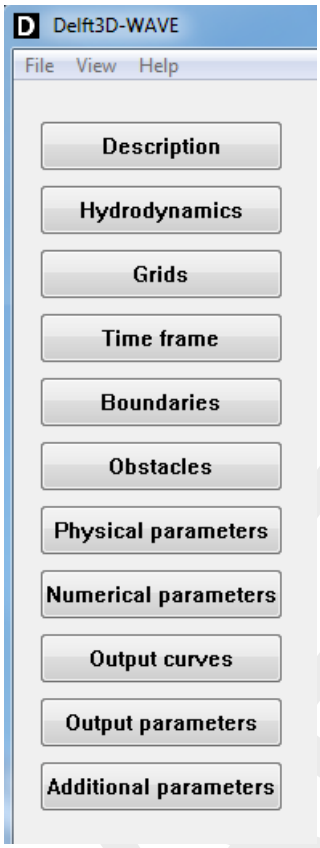

Figure 4.1: *Options in the main window of the WAVE Graphical User Interface*

4.4 Working with the WAVE-GUI

The purpose of the WAVE-GUI is to provide a graphical tool that simplifies the preparation of an MDW-file. The layout of the GUI has been shown in the figures in Chapter 3. Below, in Figure 4.1, a graphical representation of the GUI and its options shows that the main window has several buttons, each of them representing a so-called data group. A data group is a coherent set of input parameters. For instance, in the Data Group *Boundaries* you can define all incident wave conditions at the boundaries. A detailed description of the data to be entered in each data group is given in section 4.5.

Description	Identification of wave computation, run id. See section 4.5.1.
Hydrodynamics	Specification of flow results to be used as input for wave computation. See section 4.5.2.
Grids	Specification of grids and bathymetry used by wave computation (grd, enc, dep). See section 4.5.3.
Time frame	Specification of (number of) times wave computation is executed. See section 4.5.4.
Boundaries	Definition of wave incident boundaries and boundary conditions (bnd). See section 4.5.5.
Obstacles	Specification of spatial obstacles to prohibit wave propagation in space. See section 4.5.6.
Physical parameters	Specification of physical parameters. See section 4.5.7.
Numerical parameters	Specification of numerical parameters. See section 4.5.8.
Output curves	Specification of location where output is generated. See section 4.5.9.
Output parameters	Specification of output to be generated. See section 4.5.10.
Additional parameters	Specifications of parameters not yet supported by a specific window

DRAFT

in the WAVE-GUI.

Remark:

- ◇ Creation or updating of files (mdw-file as well as attribute files) requires that you save the new data immediately after their definition, or else these modifications might be lost and must be redefined.

To start the WAVE-GUI you must, in short, execute the following commands, see Chapter 3 for details:

- ◇ Click the *Delft3D-MENU* icon on the desktop (PC) or execute the command Delft3D-MENU on the command line (Linux).
- ◇ Click the menu item *Wave*.
- ◇ Change to your project or working directory.
- ◇ Click the menu item *Wave input*; the WAVE-GUI will be started and the main window will be opened.

You are now ready to start defining or modifying all input parameters, grouped into the data groups as shown in Figure 4.1.

In the menu bar you can choose from the following options:

File	For opening, saving an MDW file, or saving an MDW file with another name, or for exiting the WAVE-GUI; sub-menu items *New, Open*, *Save*, *Save As* and *Exit*, respectively.
View	For viewing the grid related parameters; sub-menu item *Visualisation Area*.
Help	For getting information on the version of the User Interface. (Note that there is no online and context help for the SWAN model available).

When leaving the WAVE-GUI you must save the mdw-file in the working directory.

4.5 Data groups of MDW-file

In this section, all input parameters in the data groups of the mdw-file will be described in the order they appear in the WAVE-GUI (see Figure 4.1).

In two of the data groups, the data is organised in sub-groups. The Data Group *Grids* is divided into the following sub-groups: *Computational grid, Bathymetry, Spectral resolution* and *Nesting*. The Data Group *Physical parameters* also consists of several sub-groups: *Constants*, *Wind*, *Processes* and *Various.*

In sub-sections 4.5.1 to 4.5.10 we will describe all data groups in consecutive order. For each input quantity we give:

- ◇ A short description of its meaning. In many cases we add a more comprehensive discussion to put the quantity and its use in perspective.
- ◇ The restrictions on its use.
- ◇ The range of allowed values, called its domain, and its default value.

DRAFT

4.5.1 Description

In the Data Group *Description* you can identify the mdw-file by giving a comprehensive description of the project, the application domain and the specific selections to be made in this scenario. The description is only used for identification and has no influence on the simulation itself. An example is displayed in Figure 4.2.

Restrictions:

- ◇ The project name may not be longer than 16 characters.
- ◇ The project number may not be longer than 4 characters.
- ◇ Three descriptive lines are allowed, each no longer than 72 characters.

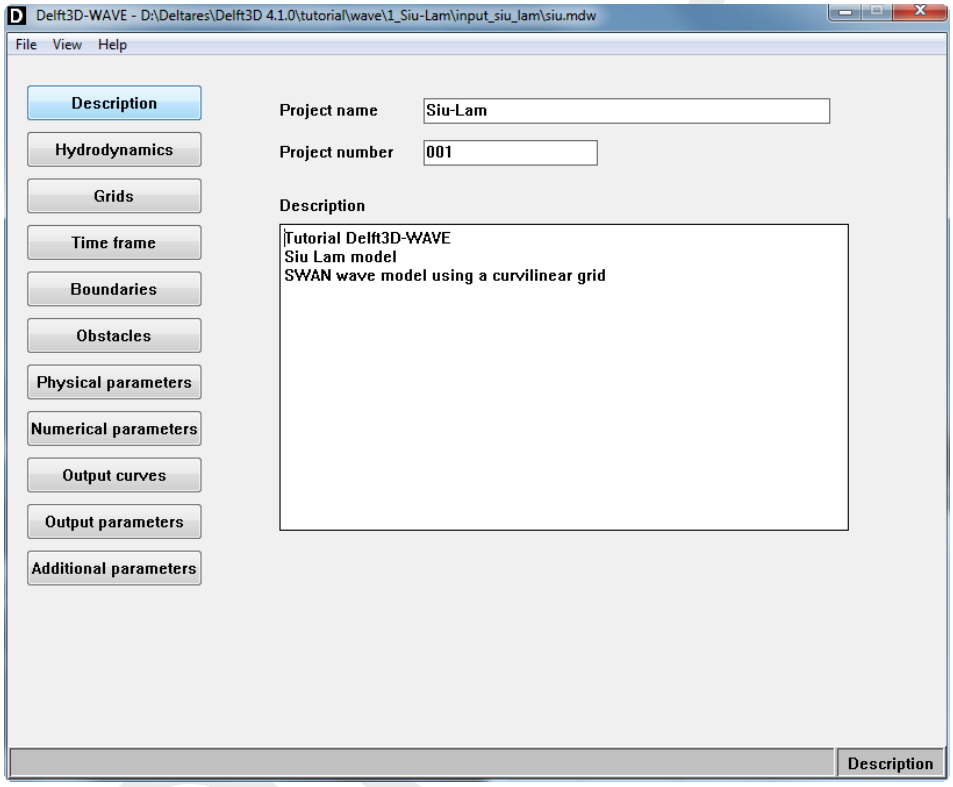

Figure 4.2: *Window of Data Group* Description

4.5.2 Hydrodynamics

As explained before in section 2.1.3, you can specify a FLOW computation from which the results are to be used as input for the wave computation (so-called offline coupling). If you want to do this, the Data Group *Hydrodynamics* is the place to define the FLOW computation to be used.

All needed results are stored in the communication file (com-file) produced by the FLOW computation (see section 2.1.3). Therefore, the FLOW com-file has to be present in your working directory.

- ◇ Click the Data Group *Hydrodynamics* to show the hydrodynamic result option, see Figure 4.3.

If you select *Use hydrodynamic result from FLOW*, the *Select FLOW file* button becomes active.

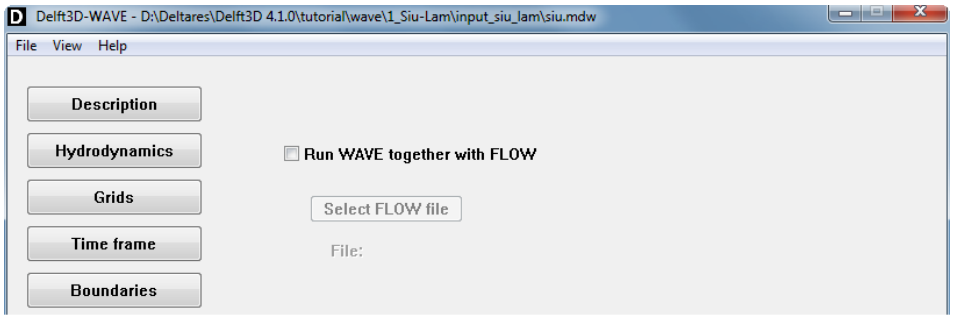

Figure 4.3: *Data Group* Hydrodynamics

By clicking *Select FLOW file* you call up a file selection menu. Select the flow computation you want to use the data from, by choosing its mdf-file (Master Definition Flow file). This is a file with the extension <mdf>.

Remarks:

- ◇ When using a FLOW model, make sure that the selected mdf-file and its associated com-file are located in your working directory, since the two modules will communicate with each other by this com-file.
- ◇ During the computations, Delft3D-WAVE determines the water depth from the bottom level, the water level and the water level correction. Bottom levels are defined as the level of the bottom relative to some horizontal datum level (e.g. a still water level), positive downward. Water levels are defined with respect to the same datum as the bottom; the water level is positive upward.

4.5.3 Grids

In this datagroup you can specify the computational grids, a computational grid is the spatial grid on which SWAN solves the wave action balance equation. In Delft3D-WAVE you can specify several grids in one run; in the tab **Nesting** you have to point out which grid is nested in which. The Data Group *Grids* consists of the following tabs:

1 **Computational grid**
One or more spatial grids on which SWAN solves the wave action balance equation.
2 **Bathymetry**
The bathymetry of the area to be modelled.
3 **Spectral resolution**
The boundaries and resolution of the directional and frequency space, which SWAN uses to perform the computations.
4 **Nesting**
When two or more computational grids are defined, you have to define which grid is nested in which.
5 **Hydrodynamics**
When results of a FLOW simulation are used, you have to specify which parameters are needed by the WAVE simulation.

DRAFT

4.5.3.1 Computational grid

You define the geographic location, size and orientation of the computational grids by importing one or more attribute grid files, *Add*, which are curvilinear grids generated with RGFGRID (grd-file). The grids can be defined in a common Cartesian co-ordinate system or in a spherical co-ordinate system, as described in Chapter 7. Once the grid is imported, the name and M and N size of the attribute grd-file are shown in the WAVE-GUI, under *Grid specifications* (see Figure 4.4).

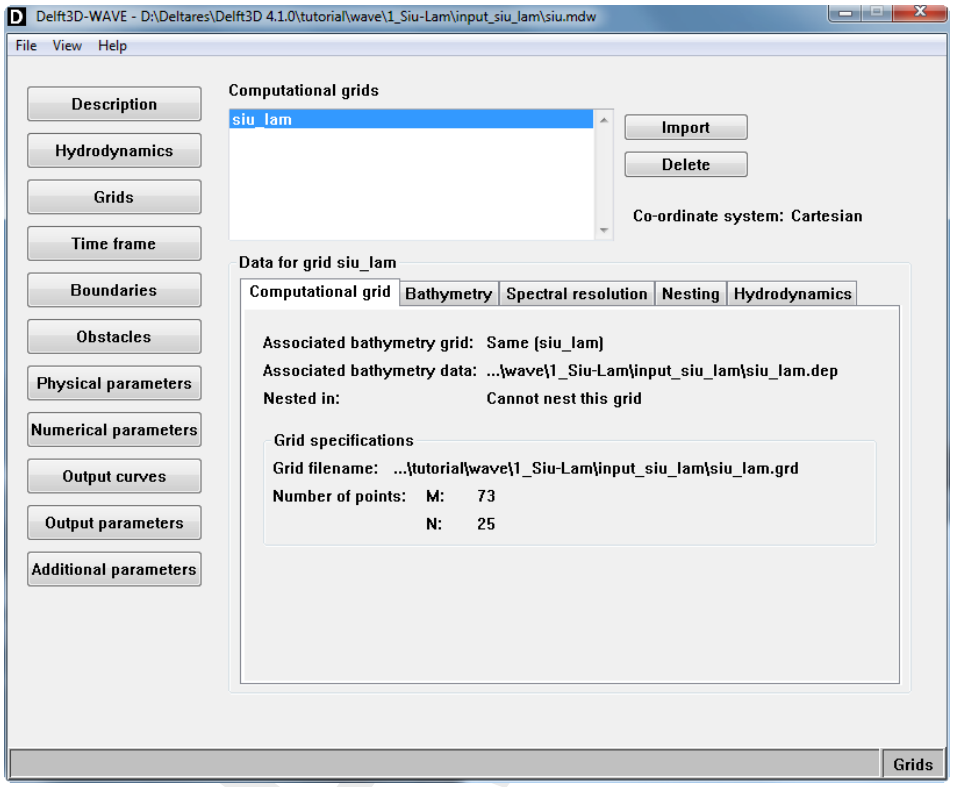

Figure 4.4: *Data Group* Grids*, sub-group* Computational grid

Remarks:

- ◇ The tab **Computational grid** also shows *Associated bathymetry grid*, *Associated bathymetry data* and *Nested in*. These data will be filled in automatically when importing the appropriate files in the tabs **Bathymetry** and **Nesting**. You are referred to the concerned sections below, for more information.
- ◇ The computational grid must be much larger than the domain where wave results are needed, because of the 'shadow' zone on both sides of the wave incident direction (see section 7.2.2).
- ◇ A grid that is created in RGFGRID always has an associated enclosure file (∗.enc). This file is not imported in the WAVE-GUI, but it will be used in case computational grids are nested, so it has to be present in the working directory.

4.5.3.2 Bathymetry

- ◇ Select the tab **Bathymetry** to work on the bathymetry of the computational grids.

As you can see in Figure 4.5, there are two ways to define the bathymetry used in the SWAN computation:

The preferable one is the first option in the WAVE-GUI: *Bathymetry data is based on* 'Com-

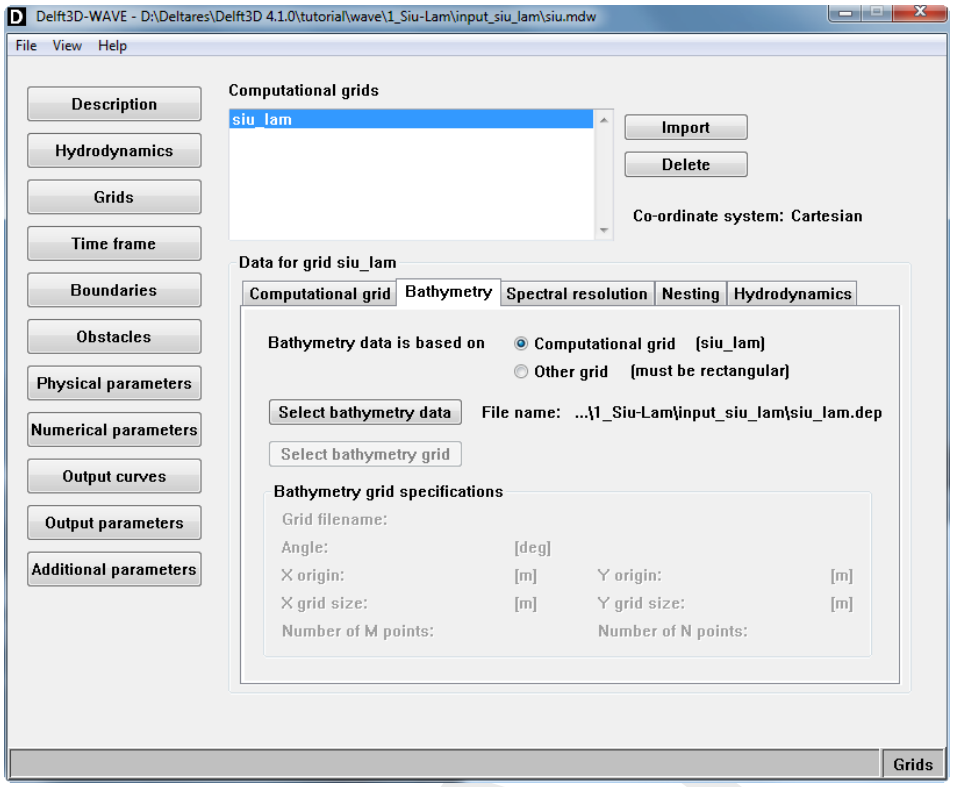

Figure 4.5: *Data Group* Grids*, sub-group* Bathymetry

putational grid'. Tick off this option. Next, you can click the button *Select bathymetry data* to import an attribute depth file (∗.dep) that is created in QUICKIN. This depth file has to be based on the computational grid (∗.grd) you imported in the tab *Computational grid*. Once the depth file is imported, the name of the file is shown in the WAVE-GUI, in both tabs **Computational grid** and **Bathymetry**.

The other option is to define the bathymetry on another, rectangular grid. This can be convenient in case you already have a rectangular grid and associated bathymetry available and you do not want to use QUICKIN to interpolate these data onto the WAVE computational grid. When you use this option, SWAN will interpolate the bathymetry data from the rectangular grid onto the computational grid, defined in the tab **Computational grid**. If you want to use this option, tick off the option *Bathymetry data is based on* 'Other grid (must be rectangular)'. Next, you have to select both the bathymetry data (∗.dep) and the bathymetry grid (∗.grd), using the buttons *Select bathymetry data* and *Select bathymetry grid*, respectively. Once the depth and grid file are imported, the names of the files are shown in the WAVE-GUI, in both tabs **Computational grid** and **Bathymetry**.

Remarks:

- ◇ In case you use the second option, where the bathymetry is based on another, rectangular bathymetry grid, the computational grid must be included strictly inside the bathymetry grid. In this way, a correct interpolation of the bathymetry data from the rectangular bathymetry grid onto the computational grid is ensured. In the region of the computational grid that lies outside the bathymetry grid, SWAN assumes that the bathymetry is identical to those at the nearest boundary of the bathymetry grid (lateral shift of that boundary). In the regions not covered by this lateral shift (i.e. in the outside quadrants of the corners of the bathymetry grid), a constant field equal to the value of the nearest corner point of the bathymetry grid is taken.
- ◇ In case you use the second option, where the bathymetry is based on another, rectan-

gular bathymetry grid, the FLOW results that you defined in the Data Group *Hydrodynamics* will first be interpolated from the FLOW computational grid onto the bathymetry grid. Next, SWAN will perform a second interpolation, where the FLOW results are transferred from the bathymetry grid to the WAVE computational grid. It is therefore sensible to ensure that the WAVE computational grid lies strictly inside the FLOW computational grid, and that the FLOW computational grid lies strictly inside the rectangular bathymetry grid. If not, no warning messages will appear, but the FLOW data will be transferred onto the bathymetry grid and SWAN computational grid with deformations.

- ◇ The formats of the depth and grid files are defined in Appendix A.

4.5.3.3 Spectral resolution

For each computational grid the spectral resolution in both directional and frequency space needs to be specified. SWAN only assigns wave energy to the wave directions and wave frequencies specified in the spectral resolution.

- ◇ Click on the **Spectral resolution** tab, see Figure 4.6.

Directional space

- ◇ *Circle*
  This option indicates that the spectral directions cover the full circle. This option is default.
- ◇ *Sector*
  This option means that only spectral wave directions in a limited directional sector are considered. The range of this sector is given by *Start direction* and *End direction*.
- ◇ *Start direction*
  This is the first direction (in degrees) of the directional sector. It can be defined either in the Cartesian or the Nautical convention (see section 7.2.1), but this has to be consistent with the convention adopted for the computation, to be defined in the Data Group *Physical parameters*.
- ◇ *End direction*
  It is the last direction of the sector (required for option Sector; Cartesian or Nautical convention, but in consistency with the convention adopted for the computation).
  Remarks:
  - ◇ The *Start direction* should be smaller than the *End direction*.
  - ◇ When *Reflections* at obstacles are activated, then the spectral directions must cover the full circle of 360°.
- ◇ *Number of directions*
  This is the number of bins in the directional space. For *Circle* this is the number of subdivisions of a full circle, so the spectral directional resolution is

$$\Delta\theta = 360^\circ / (\text{Number of directions})$$

  In the case a directional sector is used, the spectral directional resolution is

$$\Delta\theta = (\text{End direction - Start direction}) / (\text{Number of directions})$$

Frequency space

- ◇ *Lowest frequency*
  This is the lowest discrete frequency that is used in the calculation (in Hz).
- ◇ *Highest frequency*
  This is the highest discrete frequency that is used in the calculation (in Hz).
- ◇ *Number of frequency bins*
  The number of bins in frequency space is one less than the number of frequencies. It

DRAFT

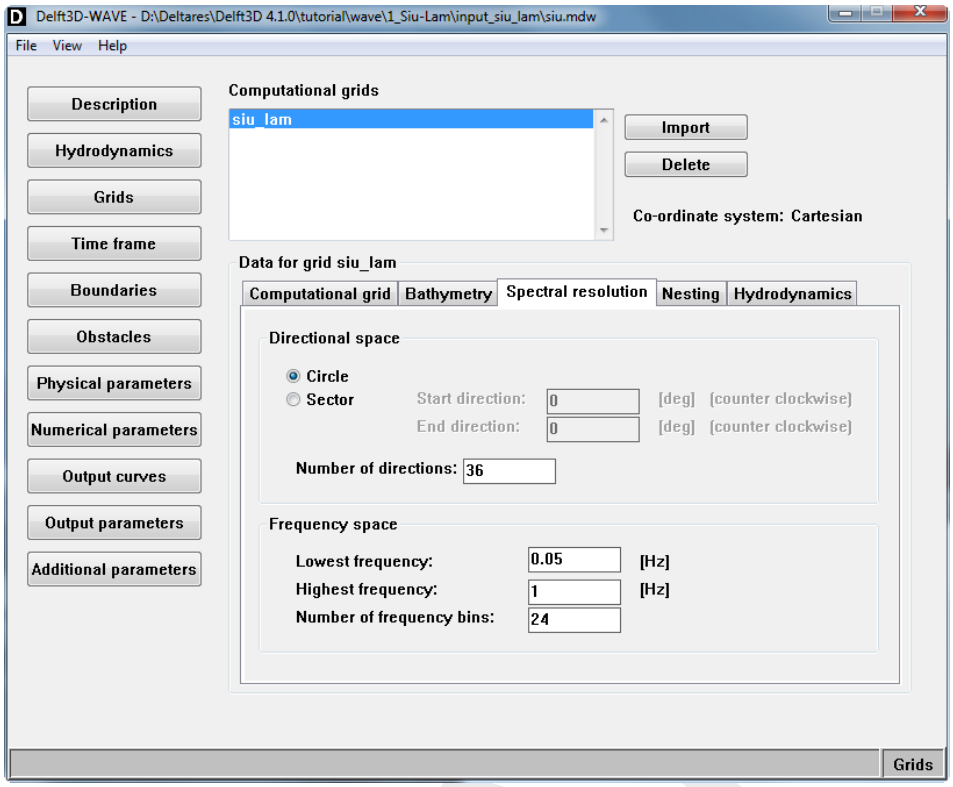

Figure 4.6: *Data Group* Grids*, sub-group* Spectral resolution

defines the resolution in frequency space between the lowest discrete frequency and the highest discrete frequency. This resolution is not constant, since the frequencies are logarithmically distributed. The number of frequency bins depends on the frequency resolution Δf that you require (see SWAN (2000), pages 39 and 49).

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Start direction	-360	360	0	degree
End direction	-360	360	0	degree
Number of directions	4	500	36	-
Lowest frequency	0.0	-	0.05	Hz
Highest frequency	0.0	-	1	Hz
Number of frequency bins	4	-	24	-

DRAFT

4.5.3.4 Nesting

Delft3D-WAVE supports the use of nested computational grids in one wave computation, See Figure 4.7. The idea of nesting is to have a coarse grid for a large area and one or more finer grids for smaller areas. The coarse grid computation is executed first and the finer grid computations use these results to determine their boundary conditions. Nesting can be repeated on ever decreasing scales.

When you want to use the nesting option, you have to import first all the computational grids and associated bathymetries as explained in the previous sub sections.

Remarks:

- The first grid cannot be nested in another one. For this grid, boundary conditions must be specified in the Data Group *Boundaries*.
- A grid cannot be nested in itself. An error message will pop up if you try this.

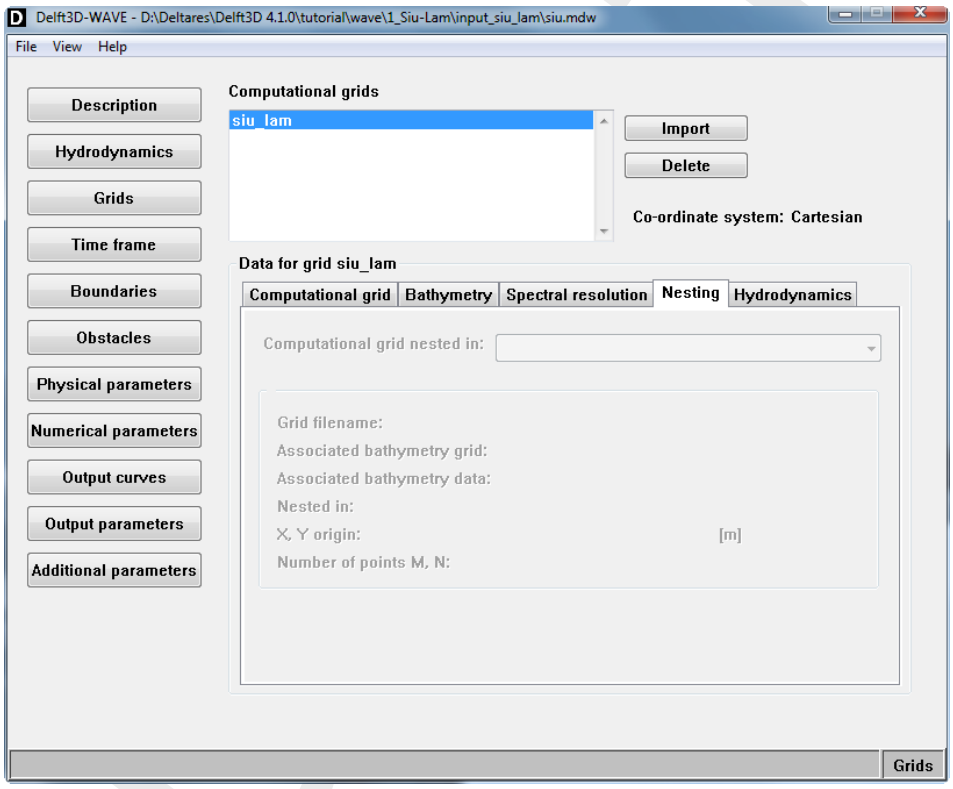

Figure 4.7: *Data Group* Grids*, sub-group* Nesting

4.5.3.5 Hydrodynamics

When the FLOW computation is performed in 2DH mode, for each of the options *Water level*, *Current*, *Bathymetry* and *Wind* the following three options can be chosen, see Figure 4.8:

- *Don't use* Don't use the quantity for the wave simulation
- *Use but don't extend* Use this quantity in the wave simulation but don't extend
- *Use and extend* Use this quantity in the wave simulation but don't extend

If the the FLOW computation is performed in 3D mode then an additional *Current type* need to be specified, see Figure 4.9. This current type can have the following values:

- *depth averaged* Use the depth averaged flow-velocity for the wave simulation.
- *surface layer* Use the flow-velocity in the surface layer for the wave simulation.

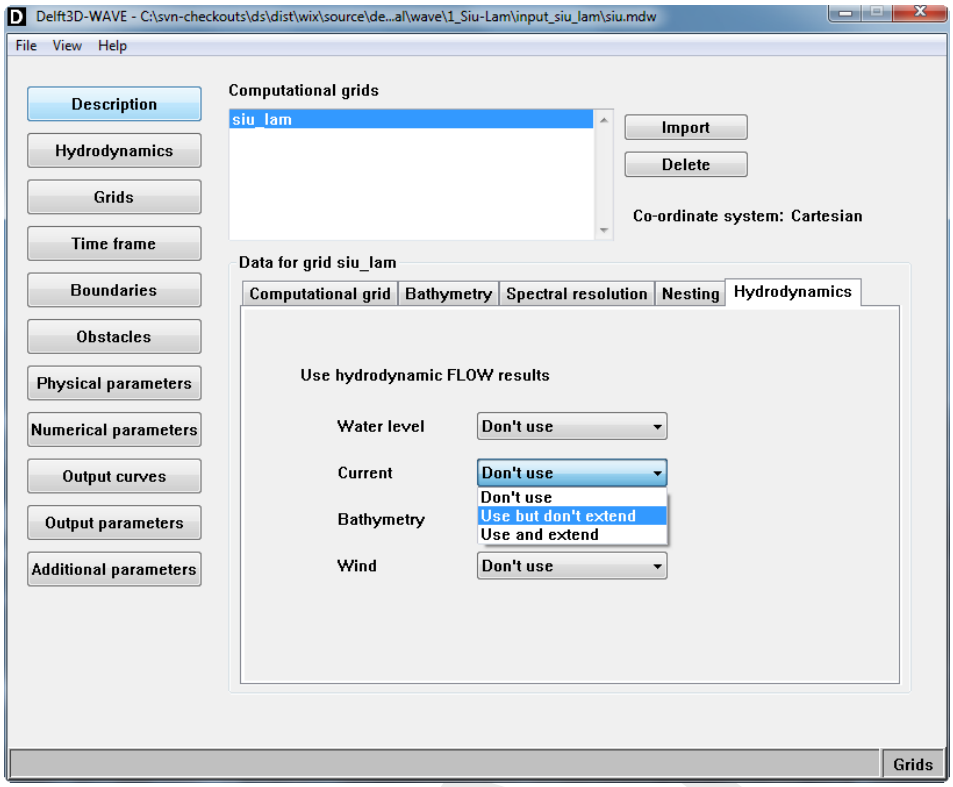

Figure 4.8: *Data Group* Grids*, sub-group* Hydrodynamics

◇ *wave dependent* A weighted flow-velocity will be used, the velocity is dependent on the orbital velocity of the wave and is especially of interest for stratified flows, see Kirby and Chen (1989).

4.5.4 Time frame

In the Data Group *Time frame*, a number of times at which wave computations must be carried out, is specified. There are three options: you want to perform a standalone wave computation, you want to perform an offline coupling with Delft3D-FLOW, or you want to perform an online coupling with Delft3D-FLOW (in the latter two cases, you specified a FLOW computation in the Data Group *Hydrodynamics*).

In all cases, in the window *Water level correction* you can specify an overall water level correction that will be applied to all water levels in the computational grid, and to all WAVE computation times specified. The water level is measured positively upward from the same datum from which bottom levels are taken. The default value is 0 m.

In the case of a coupling with Delft3D-FLOW, it can be useful to extend FLOW data on the wave grid(s) in areas that are not covered by the FLOW grid. In this way, a (more) uniform wave field can be computed at the boundaries of the FLOW grid, which can be essential during e.g. a morphological simulation.

In this window you must prescribe on which wave grid(s) you want to apply the extension of FLOW data. If no extension is required, you must choose ‘0’.

The differences between the three options in the Data Group *Time frame* are explained below.

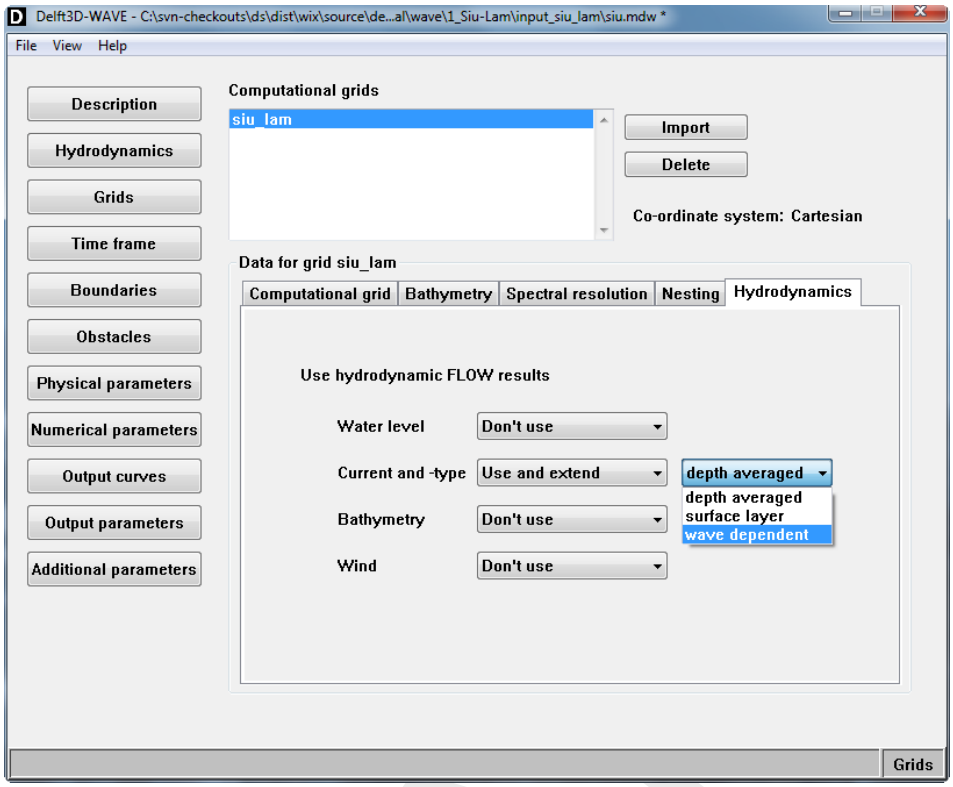

Figure 4.9: *Data Group* Grids*, sub-group* Hydrodynamics

Standalone WAVE computation

In case you want to perform a standalone WAVE computation, the Data Group *Time frame* looks like Figure 4.10.

You can add a time yourself (using the *Add* button and, if necessary, editing the time in the *Time* edit field).

Furthermore, for each WAVE computation time you can enter the following hydrodynamic properties:

- ◇ *Water level* (Default: 0 m)
  This parameter specifies a constant water level over the entire WAVE model. The water level is measured positively upward from the same datum from which bottom levels are taken.
- ◇ *X-velocity* (Default: 0 m/s)
  This parameter specifies a constant x-velocity over the entire WAVE model. The x-velocity is measured according to the Cartesian system.
- ◇ *Y-velocity* (Default: 0 m/s)
  This parameter specifies a constant y-velocity over the entire WAVE model. The y-velocity is measured according to the Cartesian system.

DRAFT

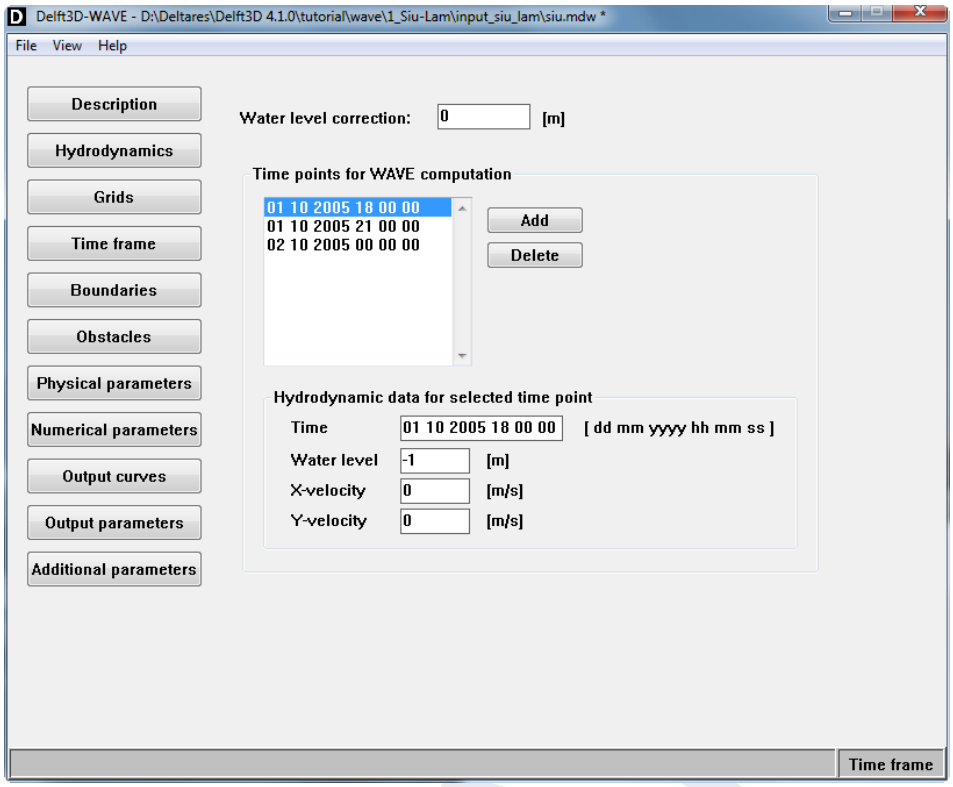

Figure 4.10: *Data Group* Time frame *in case of standalone WAVE computation*

Coupling with Delft3D-FLOW, using FLOW results

In case FLOW results have been selected in the Data Group *Hydrodynamics*, the results are read from the com-file and interpolated from the computational FLOW grid to the computational WAVE grid. Usually the FLOW grid is chosen smaller than the WAVE grid. Therefore an option is available to extend the values at the boundary of the FLOW grid to the boundary of the WAVE grid. To achieve this, the option *Extend the FLOW results on the last grid(s)* should be set to the number of grids you want to extend the FLOW results to (Figure 4.11). Furthermore, you specify which hydrodynamic results should be extended.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Time	-	-	current date 00:00:00	-
Water level	-100	100	0.	metre

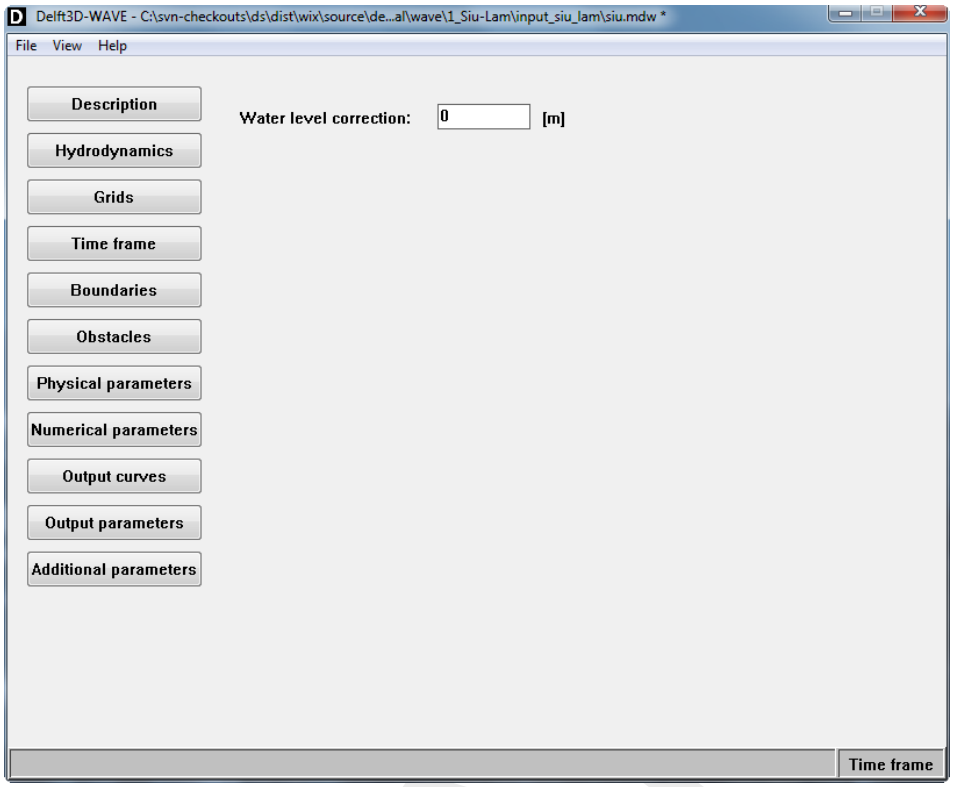

Figure 4.11: *Data Group Time frame, using FLOW results*

4.5.5 Boundaries

In the Data Group *Boundaries* the incident wave conditions at the boundary of the first and only the first, computational grid are prescribed (see Figure 4.12). All other computational grids (i.e. the nested grids) obtain their boundary information from other grids.

In the WAVE computations, wave boundary conditions may be specified at different sides. The number of sides at which boundary conditions are provided is zero by default. To specify that one or more (up to 4) boundary sides are present click *Add* and, if necessary, edit the name of the boundary in the *Boundary name* window.

The general procedure to specify boundary conditions is the following. For each of the boundaries:

1 Specify if the boundary should be defined by *Orientation*, *Grid coordinates* or *XY coordinates*.
2 Select the orientation of the boundary considered (i.e. at which direction is it located).
3 Specify if the values of the incident wave conditions are *Constant* or *Variable* along the boundary.
4 Select if the incident wave conditions are specified in terms of integral wave parameters or are read from file (with 1D or 2D wave spectra).
5 Specify the actual values of the incident wave conditions in the sub box *Edit conditions*.

Below, each of the five steps described above is explained further.

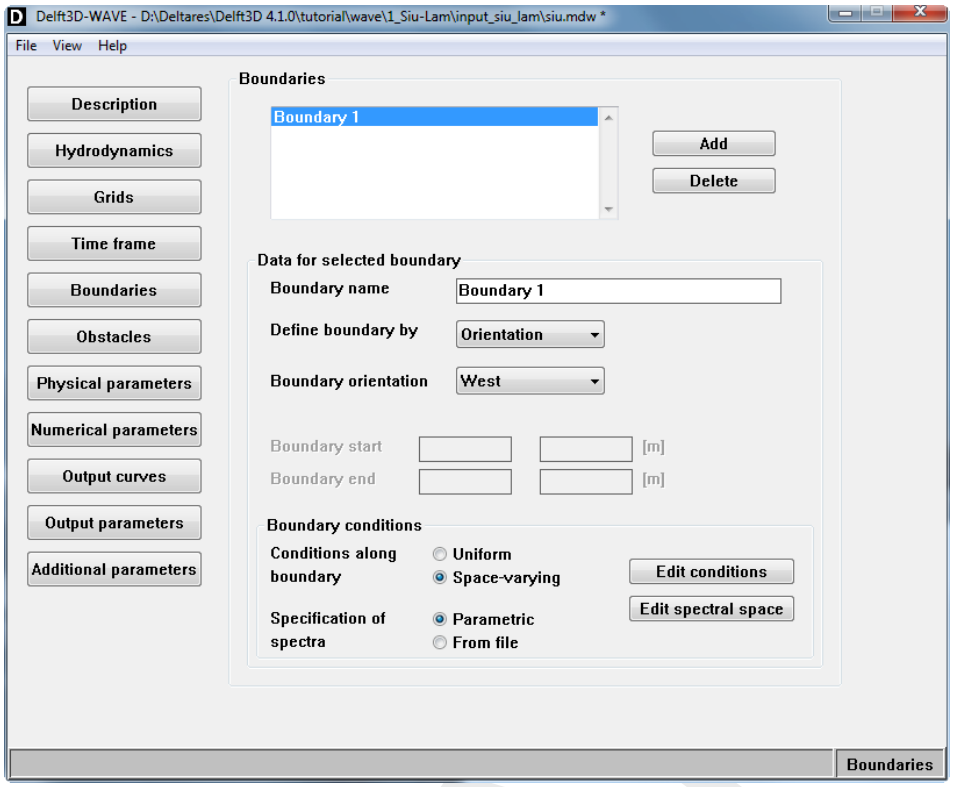

Figure 4.12: *Data Group* Boundaries

Boundary definition

There are two ways to define the boundary at which the conditions are imposed. The first (*Orientation*) is easiest if the boundary is one full side of the computational grid. The second option, i.e. segment (defined by *Grid coordinates* or *XY coordinates*) can be used if the boundary segment for instance goes around the corner of the grid, or if the segment is only part of one side of the grid.

Boundary orientation

Once you specified how you want to define a boundary in the *Define boundary by* dropdown box, you have to either enter the orientation of the boundary (in the *Boundary orientation* dropdown box) or enter *Grid coordinates* or *XY coordinates* (in the *Boundary start* and *Boundary end* input fields):

Orientation
In case the boundary is defined by its *orientation*, the boundary is considered along one full side of the computational grid. Since in the SWAN computations wave boundary conditions may be specified at 4 sides, it is necessary to indicate on which side the boundary condition is applied by selecting the orientations i.e. *North*, *Northeast,* etc. The side does not have to face exactly the given direction (the nearest direction of the normal to the side is taken; for curvilinear grids the side is taken between the first and last position of the side except when there is an interruption in the side then it is subtracted from the side). Note that the direction of the problem co-ordinate system must be defined by you, by default the positive x-axis points East.

In case the boundary is defined by *Orientation*, select the *Boundary orientation*, see Figure 4.13.

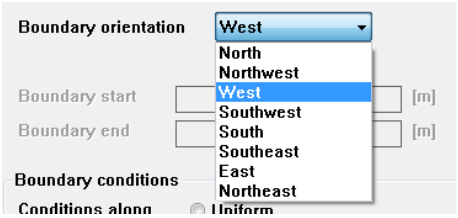

Figure 4.13: *Boundary orientations*

Co-ordinates
In case the boundary is defined by its location, either *Grid coordinates* or *XY coordinates* have to be entered. This option is used if the boundary segment goes around a corner of the grid, or if the segment is only part of one side of the grid. The distance along the segment is measured from the first point of the segment.

Specify the *Start* and *End* of the boundary in terms of *Grid* or *XY co-ordinates*, see Figure 4.14.

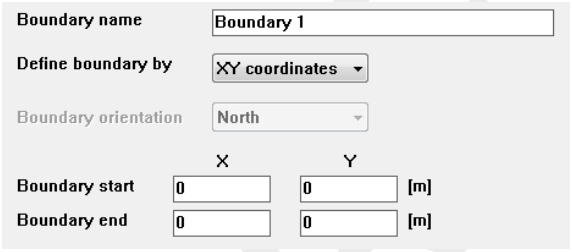

Figure 4.14: *Definition of boundary using XY coordinates*

Once you defined the names and locations of the boundaries, you can specify the boundary conditions for each boundary. First, you choose the type of *Conditions along boundary* (either uniform or space-varying), and second you tick the desired *Specification of spectra* (either parametric or from file).

Conditions along boundary

The boundary condition may be *Uniform* for a boundary side (or segment) of the first computational grid, but it may also be considered as *Space-varying:*

- *Uniform*
  With this option the wave conditions are constant along a side (or segment).
- *Space-varying*
  With this option the wave spectra can vary along the side (or segment). The incident wave field is prescribed at a number of points of the side (or segment). These points are characterised by their distance from the begin point of the side or segment. The wave spectra for grid points on the boundary of the computational grid are calculated by SWAN by the spectral interpolation.

Specification of spectra

The boundary conditions in SWAN can be specified in terms of integral wave parameters (*Parametric*) or they can be read from an external file (*From file*).

- *Parametric*
  With this option you define the boundary condition as parametric spectral input. The parameters (i.e. the spectral shape, the wave period and the directional spreading) can be

specified by clicking on the button *Edit spectral space*.

◇ *From file*
With this option the boundary condition are read from an external file (bnd-file).

Next, the actual boundary conditions can be entered in the window that appears when you click the button *Edit conditions*.

Edit conditions

The structure of the *Edit conditions* sub-window depends on the type of condition along the boundary (i.e. *Uniform* or *Space-varying*). and on the boundary specification of spectra (i.e. *Parametric* or *From file*)

If the option *Space-varying* is selected, you should also select the option *Clockwise* or *Counter clockwise*. The length along a *Side* is measured in *Clockwise* or *Counter clockwise* direction. The option counter clockwise is default. In case of a *Segment* the length is measured from the indicated begin point of the segment.

All the options are summarized in the table below, and clarified in more detail in the text below the table.

	Parametric	***From file***
Uniform	The parameters that you have to define in the sub-window *Edit conditions* are: ◇ *Significant wave height* ◇ *Wave period* ◇ *Direction* ◇ *Directional spreading*	You have to indicate the file of the boundary condition by clicking on the button *Select filename*, choosing the file in the list and adding the filename by clicking the *OK* button.
Space-varying	The parameters that you have to define in the sub-window *Conditions* are: ◇ *Distance from corner point* ◇ *Significant wave height* ◇ *Wave period* ◇ *Direction* ◇ *Directional spreading*	The parameters that have to be defined in the sub-window *Conditions* are: ◇ *Distance from corner point* Each section has to be added to the list by clicking *Add*. You have to indicate the file of the boundary condition by clicking on the button *Select filename*, choosing the file in the list and adding the filename by clicking the *OK* button.

1 *Uniform* and *Parametric*
If the *Conditions along boundary* is set to *Uniform* and the *Specification of spectra* is set to *Parametric* then the following parameters have to be specified in the window **Uniform boundary conditions**. This window will appear after pressing the button *Edit conditions*, see Figure 4.15.

◇ *Significant wave height*
The significant wave height specified in m.

◇ *Wave period*
The characteristic period of the energy spectrum. It is the value of the peak period (in

s) if option *Peak* is chosen in the *Spectral space* sub-window or it is the value of the mean period if option *Mean* is chosen in the above same sub-window.

◇ *Direction*
Mean wave direction (direction of wave vector in degree) according to the Nautical or Cartesian convention.

◇ *Directional spreading*
This is the directional standard deviation in degrees if the option *Degrees* is chosen in the **SWAN Spectral Space** window; or it is the power *m* if the option *Cosine power* is chosen in the same window.

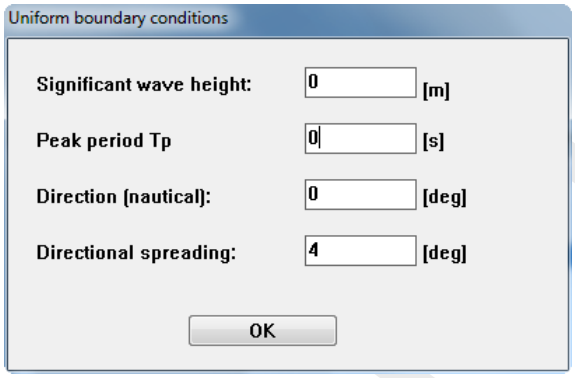

Figure 4.15: *Window* ***Uniform boundary conditions****. After pressing* Edit Conditions *when* Uniform *and* Parametric *where selected*

2 *Space-varying* and *Parametric*
If the *Conditions along boundary* is *Space-varying*, in addition to the above mentioned parameters you have to define also the *Distance from corner point*.

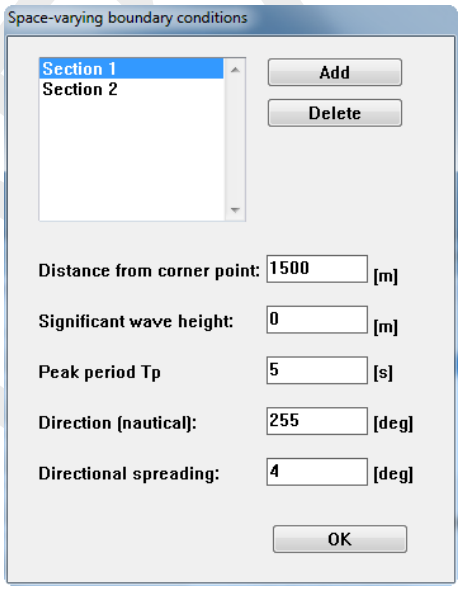

Figure 4.16: *Window* ***Space-varying boundary conditions****. After pressing* Edit Conditions *when* Space-varying *and* Parametric *where selected*

◇ *Distance from corner point*
It is the distance from the first point of the side or segment to the point along the side or segment for which the incident wave spectrum is prescribed. Note that these points do not have to coincide with grid points of the computational grid. *Distance from corner*

point is the distance in [m], not in grid steps. The values should be given in ascending order. The length along a side is measured in clockwise or counter-clockwise direction depending on the option *Wave angle* (see below). In case of a *Segment* option the length is measured from the indicated begin point of the segment.

The boundary wave spectrum at a location has to be added to the list by clicking *Add*.

3 *Uniform* and *From file*
If the *Conditions along boundary* is set to *Uniform* and in the *Boundary specification* the option *From file* is chosen, then you have to specify the filename where the input boundary spectra is located. You can specify the filename after pressing the button *Edit conditions*.

4 *Space-varying* and *From file*
If the option is *Space-varying* you have also to specify the *Distance from the corner point* (see above) and to add the section in the listbox by clicking *Add*.

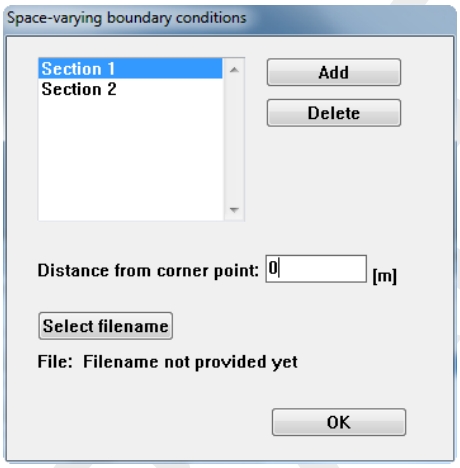

Figure 4.17: *Window* ***Space-varying boundary conditions****. After pressing* Edit Conditions *when* Space-varying *and* Parametric *where selected.*

Remark:
◇ For the correct format of the boundary file reference is made to section A.2.9.

Edit spectral space

In this sub-window you define the shape of the spectra (both in frequency and directional space) and the parameters that will be used as input, at the boundary of the first computational grid.

Shape: With this option you can define the shape of the input spectra.

◇ *JONSWAP* (default)
This option indicates that a JONSWAP type spectrum is assumed.
Peak enh. Fact.
This is the peak enhancement parameter of the JONSWAP spectrum. The default value is 3.3.

◇ *Pierson-Moskowitz*
This option means that a Pierson-Moskowitz type spectrum will be used.

◇ *Gauss*
This option indicates that a Gaussian-shaped frequency spectrum will be used. If this option is used, the width of the spectrum in frequency space has to be specified. Selecting this option the *Spreading* box will be enabled.

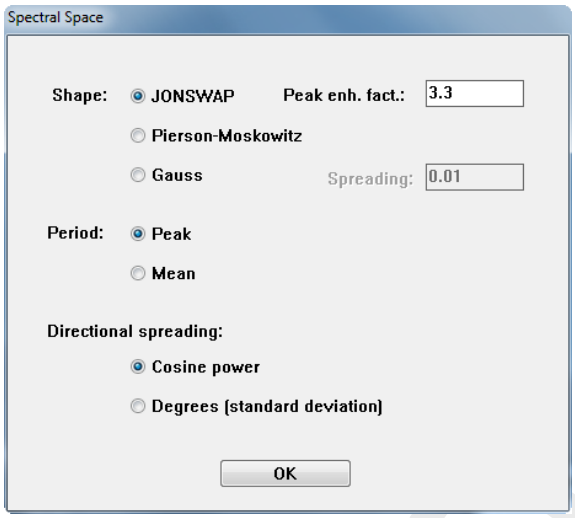

Figure 4.18: *Window* ***Space-varying boundary conditions****. After pressing* Edit spectral space *when* Space-varying *and* Parametric *where selected.*

Spreading
Width of the Gaussian frequency spectrum expressed as a standard deviation in [Hz].

Period: With this input you can specify which wave period parameter (i.e. *Peak* or *Mean* period) will be used as input.

- ◇ *Peak* (default)
  The peak period T_p is used as characteristic wave period.
- ◇ *Mean*
  The mean wave period T_{m01} is used as characteristic wave period. For the definition see Appendix B.

Directional spreading: With this input you can specify the width of the directional distribution. The distribution function itself is: $\cos(\theta - \theta_{peak})$.

- ◇ *Cosine power* (default)
  The directional width is expressed with the power m itself.
- ◇ *Degrees (standard deviation)*
  The directional spreading is expressed in terms of the directional standard deviation of the $[\cos(\theta - \theta_{peak})]$ distribution (for a definition see Appendix B).

In case the boundary conditions are to be read from file then select *From file*:

- ◇ *From file*
  This option means that the boundary is read from an external file in which the spectra at the boundary are specified (note that only the incoming wave components of these spectra are used by SWAN).

DRAFT

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Number of points to specify boundary	0	300	0	-
Spectral peak factor	1.	10.	3.3.	-
Distance from corner point	0.	Y-length	0.	m
Significant wave height	0.	25.	0.	m
Spectral peak period	0.1	20.	1.	s
Wave direction	-360	360.	0.	°
Directional width (m)	1.	100.	4.	-

4.5.6 Obstacles

Within the Data Group *Obstacles* you can specify the characteristics of a (line of) sub-grid obstacles through which waves are transmitted or against which waves are reflected or both at the same time (see Figure 4.16). The location of the obstacle is defined by a sequence of corner points of a polyline. The obstacles interrupt the propagation of the waves from one grid point to the next wherever this obstacle line is located between two neighbouring grid points of the computational grid (the resolution of transmission or blockage is therefore equal to the computational grid spacing).

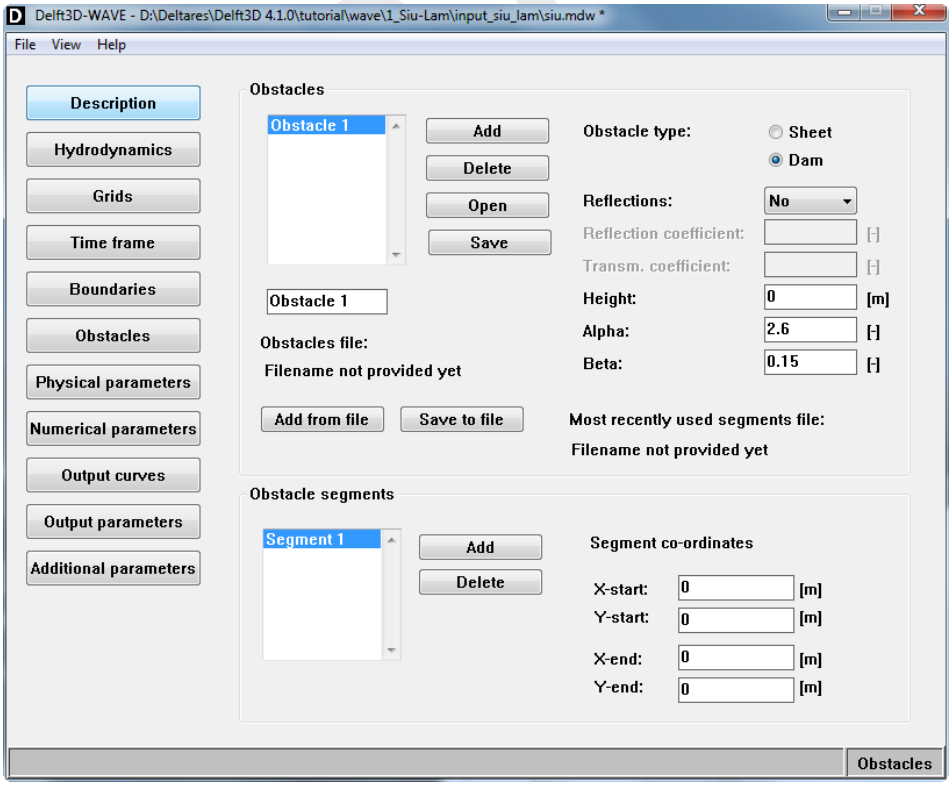

Figure 4.19: *Data Group* Obstacles

By clicking *Add* you specify that — at least — one obstacle is present (the button *Add* may

be used more than once to define more obstacles). For this obstacle, you should specify the type of the obstacle and the co-ordinates of the corner points. Use button *Delete* to delete an obstacle, use button *Open* to open and read an obstacle file and button *Save* to save an obstacle file

With respect to the type of the obstacle, the following options are available:

- ◇ *Sheet*: With this option you indicate that the transmission coefficient is a constant along the obstacle.
- ◇ *Dam*: With this option you indicate that the transmission coefficient depends on the incident wave conditions at the obstacle and on the obstacle height (which may be submerged).
- ◇ *Reflections*: With this option you can specify if the obstacle is reflective (specular or diffusive; possibly in combination with transmission) and the constant reflection coefficient.
- ◇ *Reflection coefficient* (default = 0)
  The reflection coefficient is formulated in terms of ratio of reflected significant wave height over incoming significant wave height.
- ◇ *Transmission coefficient* (default = 1.0)
  is the transmission coefficient for the significant wave height (coefficient = 0.0: no transmission = complete blockage).
- ◇ *Height* (default = 0.0)
  The elevation of the top of the obstacle above the reference level (same reference level as for bottom etc.); use a negative value if the top is below that reference level (possibly in case of submerged obstacles).
- ◇ *Alpha* (default = 2.6)
  Coefficient determining the transmission coefficient depending on the shape of the dam (see section 7.3.2).
- ◇ *Beta* (default = 0.15)
  Coefficient determining the transmission coefficient depending on the shape of the dam (see section 7.3.2).
- ◇ *Add from file* Load an extra obstacle segment file, for the file format see section A.2.5.
- ◇ *Save to file* Save obstacle segments to file which is listed below the text *Most recently used segments file:*

Remark:

- ◇ When *Reflections* at obstacles are activated, then for each computational grid the directional space should be *Circle* or *Sector* covering the full circle of 360°.

Once it has been determined which type of obstacle is used, the location of the obstacle must be specified by the co-ordinates of the corner points of the obstacle (at least two corner points must be provided).

The *X-start* and *Y-start* co-ordinates represent the location of the first corner point of the obstacle. The next set of co-ordinates must be given in the *X-end* and *Y-end* co-ordinate boxes. Adding one extra set of co-ordinates is equal to adding one segment to the obstacle.

When a lot of obstacles have to be defined, the procedure described above can be quite cumbersome. Therefore, it also possible to define a number of obstacles by importing a polyline file in which you defined the corner points of the obstacles. This is done by clicking on the button *Open*.

Remarks:

- ◇ Reflections will only be computed if the spectral directions cover the full 360°.
- ◇ In case of specular reflection the angle of reflection equals the angle of incidence.

DRAFT

◇ In case of diffuse and scattered reflection in which the angle of reflection does not equal the equal the angle of incidence.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Reflection			**No**	
Reflection coefficient	0.0	1.0	0.0	-
Sheet (max number = 250):				
Transmission coefficient	0	1	1.0	-
Dam (max number = 250):	.			
Height	-100.	+100.	0.	m
Alpha	1.8	2.6	2.6	-
Beta	0.1	0.15	0.15	-

4.5.7 Physical parameters

In the Data Group *Physical parameters* you may specify a number of physical parameters. The following options are possible (see Figure 4.20):

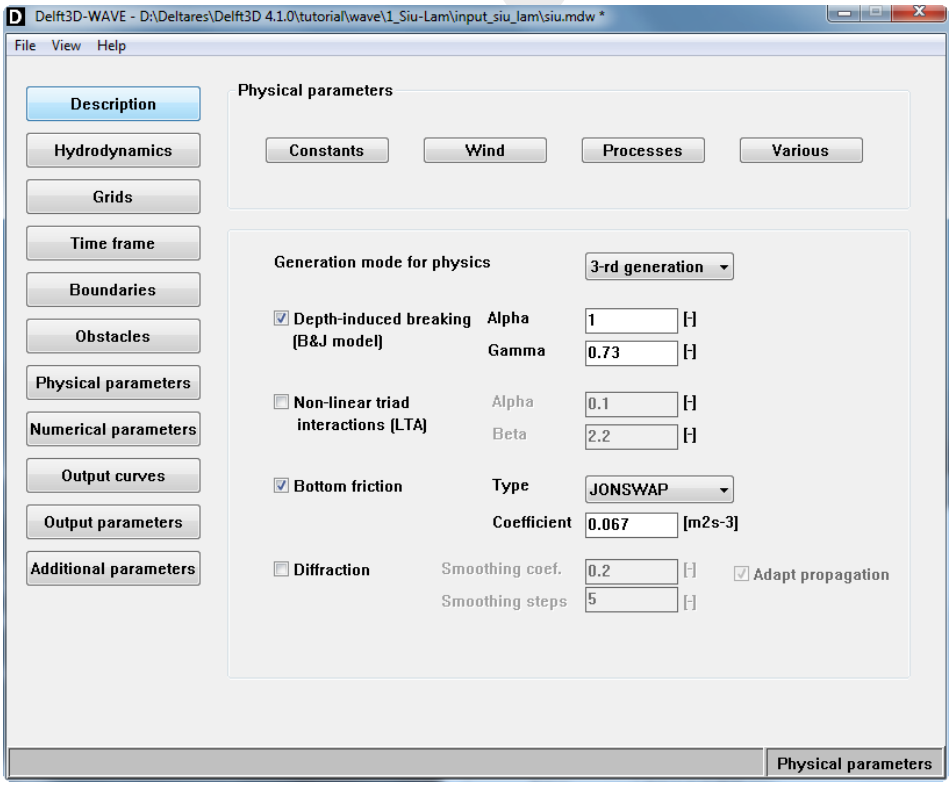

Figure 4.20: *Data Group* Physical parameters

◇ *Constants*: Within this sub-data group you can assign values to some parameters.

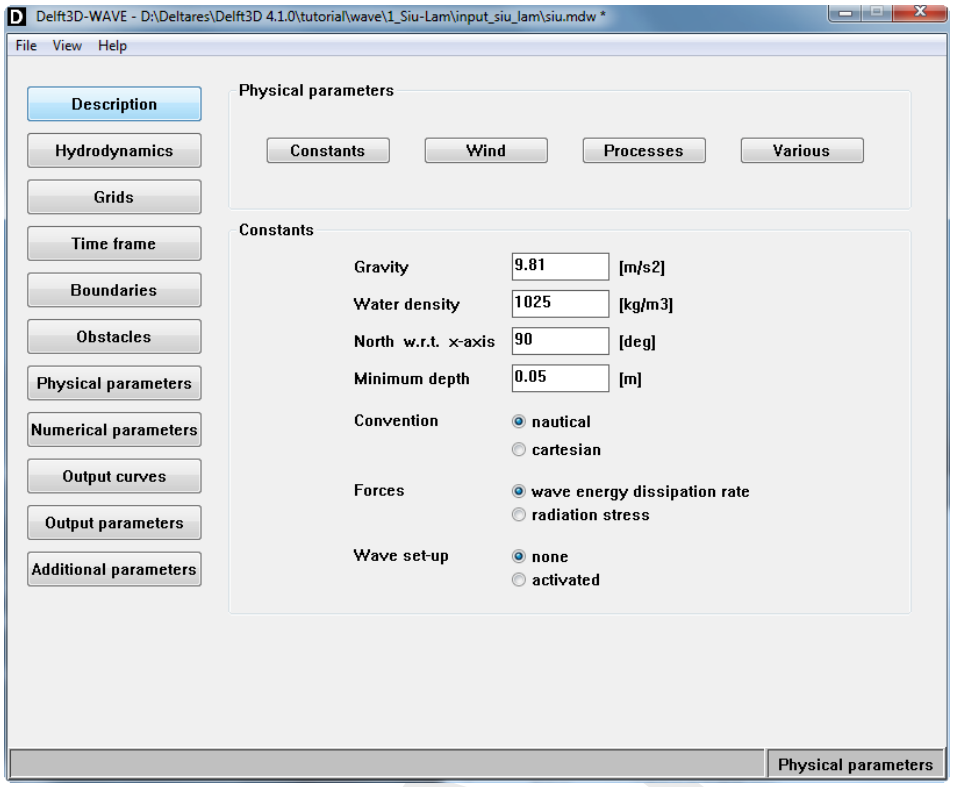

Figure 4.21: *Sub-data Group* Physical parameters, Constants

- ◇ *Wind*: Here you can specify the wind conditions (for a standalone simulation).
- ◇ *Processes*: With these parameters you can influence some of the physical processes of SWAN (i.e. type of formulation, dissipation processes, non-linear wave-wave interactions).
- ◇ *Various*: With these parameters you can influence the wave propagation in the spectral space and the physical processes in SWAN.

Remark:

- ◇ If the wind parameters are used from the FLOW computation, the Sub-data Group *Wind* is invisible.

4.5.7.1 Constants

In the Sub-data Group *Constants* you can specify the following parameters (see Figure 4.21):

Gravity
The gravitational acceleration in m/s^2. The default value is 9.81 m/s^2.

Water density
The water density ρ in kg/m^3. The default value is 1025 kg/m^3.

North
The direction of North with respect to the x-axis (Cartesian convention). The default value is 90° i.e. x-axis pointing East.

Minimum depth
The threshold depth in [m]; in the computation any positive depth smaller than this threshold depth is set to the threshold depth. The default 0.05 m.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Acceleration of gravity	9.8	10.	9.81	m/s^2
Density of water	950.	1050.	1025.	kg/m^3
North	-360.	360.	90.	deg
Minimum depth	-	-	0.05	m

Convention
In the input and output of SWAN the direction of wind and waves are defined according to either the *Cartesian* convention or the *Nautical* convention (see Figure 7.1 for definitions).

- ◇ *Cartesian*
  This option indicates that the Cartesian convention for wind and wave direction (SWAN input and output) will be used. The direction is the angle between the vector and the positive x-axis, measured counter-clockwise (the direction where the waves are going to or where the wind is blowing to).
- ◇ *Nautical*
  This option indicates that the nautical convention for wind and wave direction will be used. The direction of the vector from the geographic North measured clockwise + 180°. This is the direction where the waves are coming from or where the wind is blowing from.

Wave set-up
If this option is activated, the wave induced set-up is computed and accounted for in the wave computations (during the computation it is added to the depth that is obtained from the bottom and the water level). This option should only be used if SWAN is applied as standalone model or if wave-induced set-up is not accounted for in the flow computations.

Forces
With the integration of the *fully spectral* SWAN model under the Delft3D model it is possible to compute the wave forces on the basis of the energy wave dissipation rate or on the gradient of the radiation stress tensor (SWAN, 2000).

4.5.7.2 Wind

If you use the wind from a FLOW simulation (both online and offline) then the *Wind* sub-data group is not visible.

In the Sub-data Group *Wind* you can specify the type of wind conditions, i.e. *uniform wind* or *space-varying wind* (see Figure 4.22).

- ◇ *Uniform Wind*:
  Wind Speed (Default: 0 m/s)
  Wind velocity at 10 m elevation (m/s).
  Wind Direction (Default: 0°)
  Wind direction at 10 m elevation (direction of wind vector in degree) according to the convention, specified in the Sub-data group *Constants*.
- ◇ Spatially varying wind can be used as a special feature in Delft3D-WAVE. It is not yet available in the WAVE-GUI. If a space-varying wind field is applied, you should specify the file(s) with the data of the wind field (x-components and y-components). The wind grid

DRAFT

can be identical to the bathymetry grid or it can be different. See section A.2.10 for details on specifying space-varying wind.

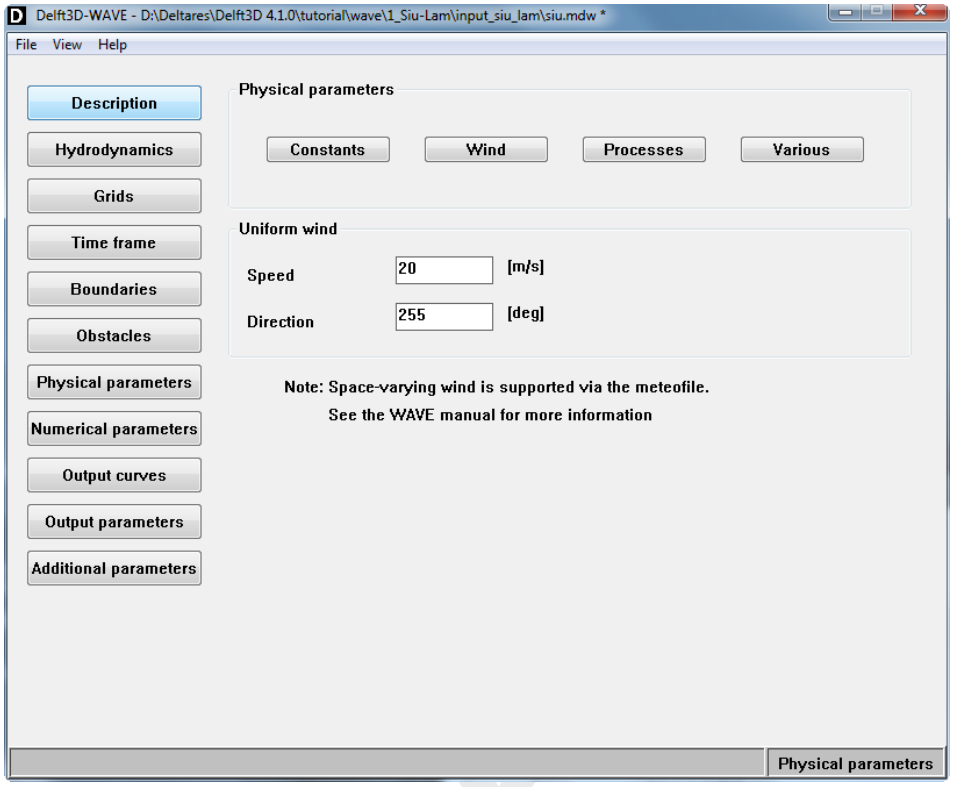

Figure 4.22: *Sub-data Group* Physical parameters, Wind

If a uniform wind speed and wind direction are applied, you should specify these values in the two boxes that are available:

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Wind speed	0.0	50.0	0.0	m/s
Wind direction	-360.0	360.0	0.0	deg

Remark:

- ◇ If the wind speed is larger than zero, and in Sub-data Group *Processes* the third generation mode is selected, then the Quadruplets in Sub-data Group *Various* will be activated.

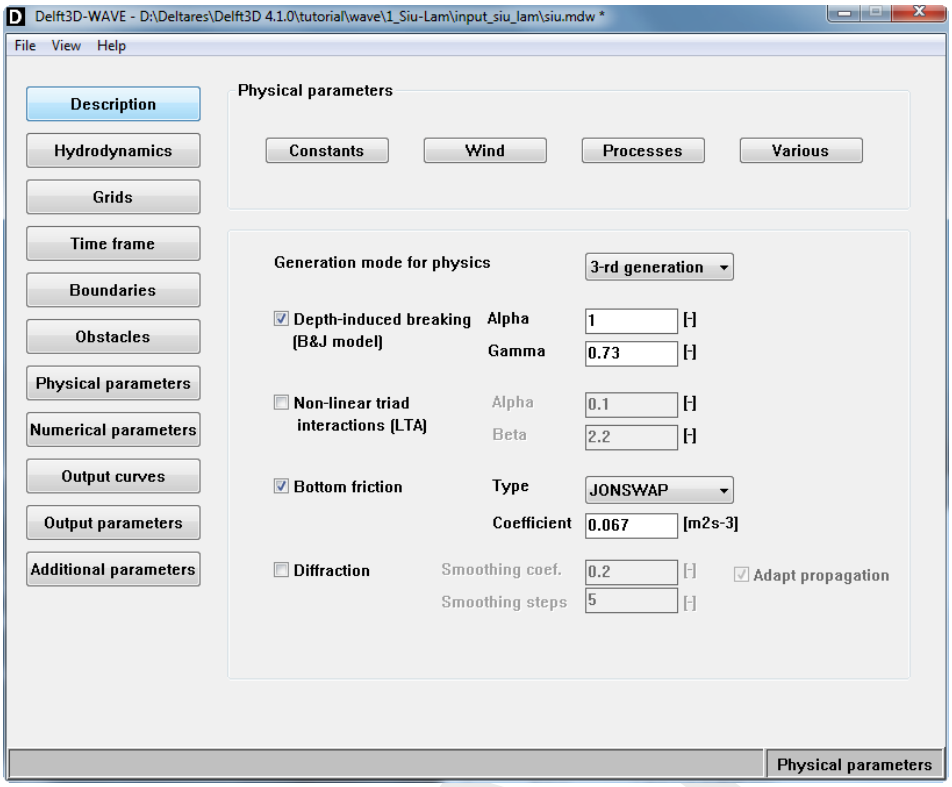

Figure 4.23: *Sub-data Group* Physical parameters, Processes

4.5.7.3 Processes

SWAN contains a number of physical processes (see Figure 4.23) that add or withdraw wave energy to or from the wave field. The processes included are: wave growth by wind, white-capping, bottom friction, depth induced wave breaking, non-linear wave-wave interactions (quadruplets and triads). SWAN can run in several modes, indicating the level of parameterisation.

◇ *Generation mode for physical formulations*:
1st generation
With this option you indicate that SWAN should run in first-generation mode.
2nd generation
With this option you indicate that SWAN should run in second-generation mode (for more information, reference is made to the SWAN manual).
3rd generation
With this option you indicate that SWAN should run in third-generation mode. Activated are wind input, quadruplet interactions and white-capping. Triads, bottom friction and depth-induced breaking are not activated by this option.

Remark:

◇ If SWAN runs in third generation mode and the wind speed is larger than zero, then the Quadruplets in Sub-data Group *Various* will be activated.

None
With this option you indicate that no deep water physical processes (i.e. wind, white-capping and quadruplets) are activated.

◇ *Depth-induced breaking*
With this option you can influence depth-induced wave breaking in shallow water in the SWAN model (see section 7.3.1). Ticking off this depth-induced term is usually unwise, since this leads to unacceptably high wave heights near beaches (the compute wave heights 'explode' due to shoaling effects).

B&J model
This option means that to model the energy dissipation in random waves due to depth-induced breaking, the bore-based model of Battjes and Janssen (1978) is used. In this option a constant breaker parameter is to be used.
Alpha
The coefficient for determining the rate of dissipation. Default = 1.0.
Gamma
The value of the breaker parameter defined as H_m/d. Default = 0.73.

◇ *Non-linear triad interactions (LTA)*
With this option you can activate the triad wave-wave interactions in the SWAN model (see section 7.3.1). Ticking off this feature means that the non-linear wave-wave interactions due to the triads are not taken into account. *LTA* means that the Lumped Triad Approximation (LTA) of Eldeberky and Battjes (1996) is used.
Alpha
The value of the proportionality coefficient α_{EB}. The default value is equal to 0.1.
Beta
This controls the maximum frequency that is considered in the computations. The value determines the ratio of the maximum frequency over the mean frequency, for which the interactions are computed. The default value is 2.2.

◇ *Bottom friction*
With this option you can activate bottom friction (see section 7.3.1). If this option is not used, SWAN will not account for bottom friction. In SWAN three different formulations are available, i.e. that of Hasselmann *et al.* (1973) (*JONSWAP*), Collins (1972); Madsen *et al.* (1988)). The default option is *de-activated*.
JONSWAP
This indicates that the semi-empirical expression derived from the JONSWAP results for bottom friction dissipation (Hasselmann *et al.*, 1973) will be activated.

- *Coefficient*
  The coefficient of the JONSWAP formulation. It is equal to 0.067 m^2s^{-3} for wind sea conditions (default value) and equal to 0.038 m^2s^{-3} for swell conditions.

Collins
This indicates that the expression of Collins (1972) will be activated.

- *Coefficient*
  The Collins bottom friction coefficient, default = 0.015.

Madsen et al.
This indicates that the expression of Madsen *et al.* (1988) is activated.

- *Coefficient*
  The equivalent roughness length scale of the bottom. Default = 0.05 m.

◇ *Diffraction*
With this option you can activate diffraction in the wave computation. The default option is de-activated. The diffraction implemented in SWAN is based on a phase-decoupled refraction-diffraction approximation (Holthuijsen *et al.*, 1993). It is expressed in terms of the directional turning rate of the individual wave components in the 2D wave spectrum. The approximation is based on the mild-slope equation for refraction and diffraction, omitting phase information.
Smoothing coefficient
During every smoothing step all grid points exchange [smoothing coefficient] times the energy with their neighbours. Default = 0.2.
Smoothing steps
Number of smoothing steps. The default value is equal to 5.
Adapt propagation

DRAFT

Switch to turn on or off the adaption of propagation of velocities in geographic space due to diffraction. The default value is activated (when diffraction is activated).

Remark:

- ◇ The process diffraction can only be solved accurately when a detailed grid is applied. Several studies (e.g. Ilic (1994)) have shown that the grid size should be about $1/10$ of the wave length; so, $dx = L/10$. In case of much coarser grids, the SWAN computation can become unstable and results are not reliable. So, use diffraction with care!

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Generation mode			**3rd generation**	
Depth-induced breaking:			**B&J model**	
Alfa	0.1	10	1.0	-
Gamma	0.55	1.2	0.73	-
Non-linear triad interactions			**inactive**	
Alfa	0.001	10	0.10	-
Beta	0.001	10	2.2	-
Bottom friction			**JONSWAP**	
Bottom friction coefficient			0.067	m^2/s^3
Diffraction			**inactive**	
Smoothing coefficient	0	1.0	0.2	-
Smoothing steps	1	999	5	-
Adapt propation			active	

4.5.7.4 Various

In the Sub-data Group *Various* some of the physical processes of SWAN (i.e. *Wind growth*, *Whitecapping*, *Quadruplets*, *Refraction* and *Frequency shift*) may be modified by you.

For initial SWAN runs, it is strongly advised to use the default values as shown in Figure 4.24. First it should be determined whether or not a certain physical process is relevant to the result. If this cannot be decided by means of a simple hand computation, you can perform a SWAN computation without and with the physical process included in the computations, in the latter case using the standard values chosen in SWAN.

For the white capping two model descriptions are possible:

1 Komen *et al.* (1984)
2 Van der Westhuysen (2007)

DRAFT

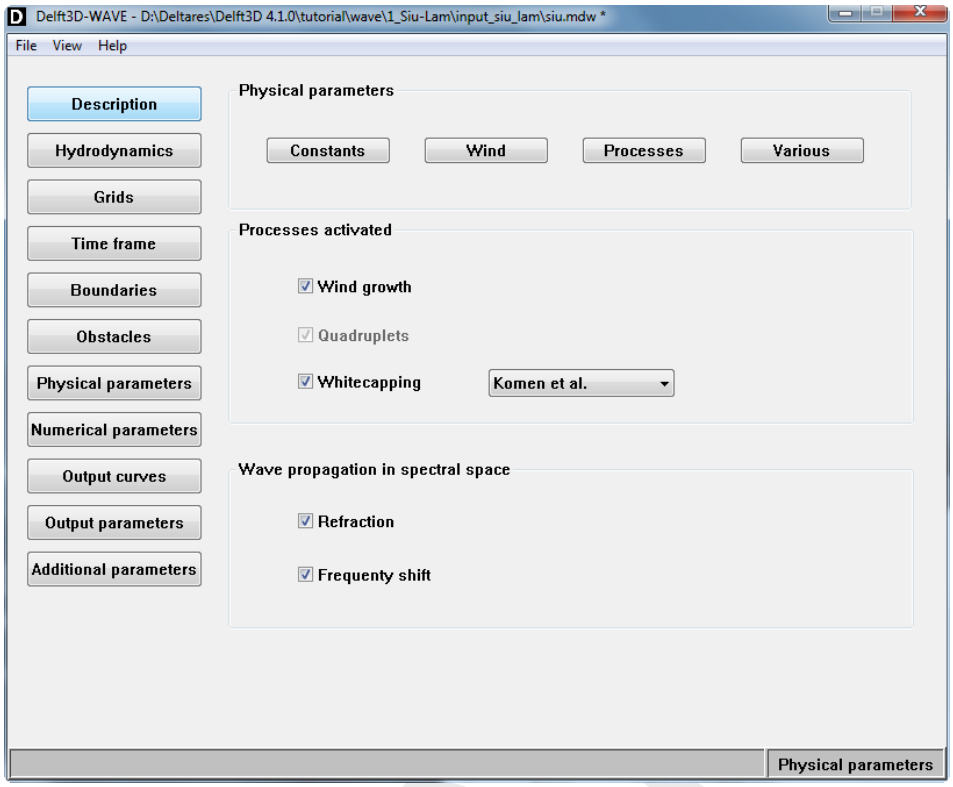

Figure 4.24: *Sub-data Group* Physical parameters, Various

Remark:

- ◇ If the wind speed is larger than zero, and in Sub-data Group *Processes* the third generation mode is selected, then the Quadruplets in Sub-data Group *Various* will be activated.

4.5.8 Numerical parameters

In the Data Group *Numerical parameters* you can modify parameters that affect the stability and accuracy of the numerical computation (see Figure 4.25). To obtain robust results with acceptable accuracy, apply the default diffusion parameters.

- ◇ *Spectral space*
  In this sub-window you can control the amount of diffusion of the implicit scheme in the directional space through the *Directional space (CDD)* parameter and frequency space through the *Frequency space (CSS)*.
  Directional space
  A value of CDD = 0 corresponds to a central scheme and has the largest accuracy (diffusion ≈ 0) but the computation may more easily generate spurious fluctuations. A value of CDD = 1 corresponds to an upwind scheme and it is more diffusive and therefore preferable if (strong) gradients in depth or current are present. The default value is CDD = 0.5.
  Frequency space
  A value of CSS = 0 corresponds to a central scheme and has the largest accuracy (diffusion ≈ 0) but the computation may more easily generate spurious fluctuations. A value of CSS = 1 corresponds to an upwind scheme and it is more diffusive and therefore preferable if (strong) gradients in current are present. The default value is CSS = 0.5.
- ◇ *Accuracy criteria (to terminate the iterative computations)*
  With these options you can influence the criteria for terminating the iterative procedure

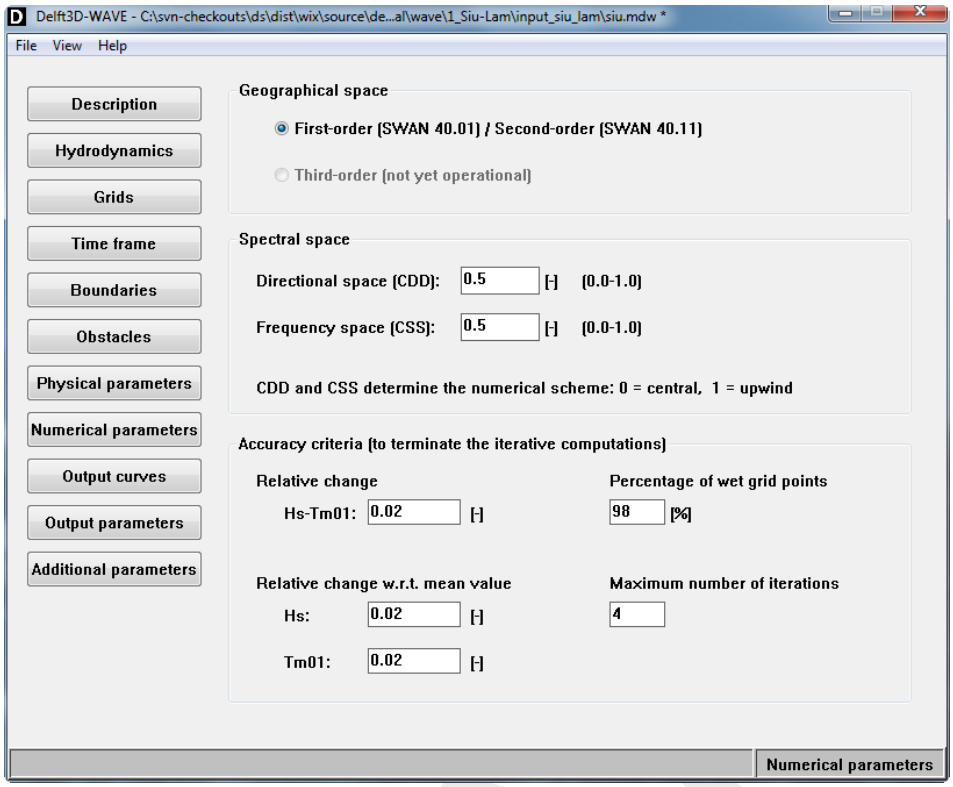

Figure 4.25: *Data Group* Numerical parameters

in the SWAN computation (for convergence criteria of SWAN see section 7.5.1). SWAN stops the iteration if:

a) The change in the local significant wave height (Hs) from one iteration to the next is less than:
   - fraction *Relative change* of that wave height or
   - fraction *Relative change w.r.t. mean value* of the average significant wave height (averaged over all wet grid points)

b) and if the change in the local mean wave period from one iteration to the next is less than:
   - fraction *Relative change* of that period or
   - fraction *Relative change w.r.t. mean value* of the average mean wave period (averaged over all wet grid points)

c) and if the conditions a) and b) are fulfilled in more than fraction *Percentage of wet grid points %* of all wet grid points.

Relative change
The default value is 0.02.
Relative change w.r.t. mean value
The default value is 0.02, for both H_s and T_{m01}.
Percentage of wet grid points
The default value is 98%.
You can also control the terminating procedure by giving the maximum number of iterations *Max. number of iterations* after which the computation stops.
Max. number of iterations
The default value is 15.

Domain:

Parameter	Lower limit	Upper limit	Default	Unit
Diffusion θ-space (directional)	0.	1.	0.5	-
Diffusion σ-space (frequency)	0.	1.	0.5	-
Relative change	0.	-	0.02	-
Relative change w.r.t. mean value (H_s and T_{m01})	0.	-	0.02	-
Percentage of wet grid points	0.	100%	98%	-
Max. number of iterations	1	-	15	-

4.5.9 Output curves

Within the Data Group *Output curves* you can specify a (curved) output curve at which wave output should be generated by Delft3D-WAVE (see Figure 4.26). Actually this curve is a broken line, defined by you in terms of segments. The values of the output quantities along the curve are interpolated from the computational grid.

By clicking *Add* in the **Output curves** canvas you add an output curve. For this output curve, you may define several segments in the **Curve segments** canvas. Each segment is defined by the co-ordinates of the begin and end points (see boxes under *Segment co-ordinates*). If you add another segment to a selected curve, the begin point of this new segment will be the end point of the previous segment. Thus you only need to specify the end point. Per segment you can specify the *Number of output stretches* along that segment. Output will be generated at equidistant locations along each segment. The total number of output locations per curve will be the sum of the *Number of output stretches* per segment plus 1.

To remove a curve with all its segments, select the curve in the **Output curves** window and click *Delete* in the same window.

To remove segments from a curve, select the segment in the **Curve segments** window and click *Delete* in the same window.

Remark:

- ◇ The names of output curves and/or curve segments as displayed in the listboxes, are not input for SWAN. The names are only displayed for your convenience. Moreover, the number in the names does not determine the sequence. The first curve in the list is the first curve specified, the second curve in the list is the second curve specified, though the name may suggest differently. Reloading this scenario will renumber the names of curves and segments but not the order.

The following output quantities will be generated by Delft3D-WAVE at the output locations along the curve.

XP, YP	co-ordinates of output location (with respect to the problem co-ordinates)
DIST	distance along the output curve (m)
DEPT	depth (in m)
HSIG	significant wave height (in m)
PER	mean wave period (T_{m01}) in s

DRAFT

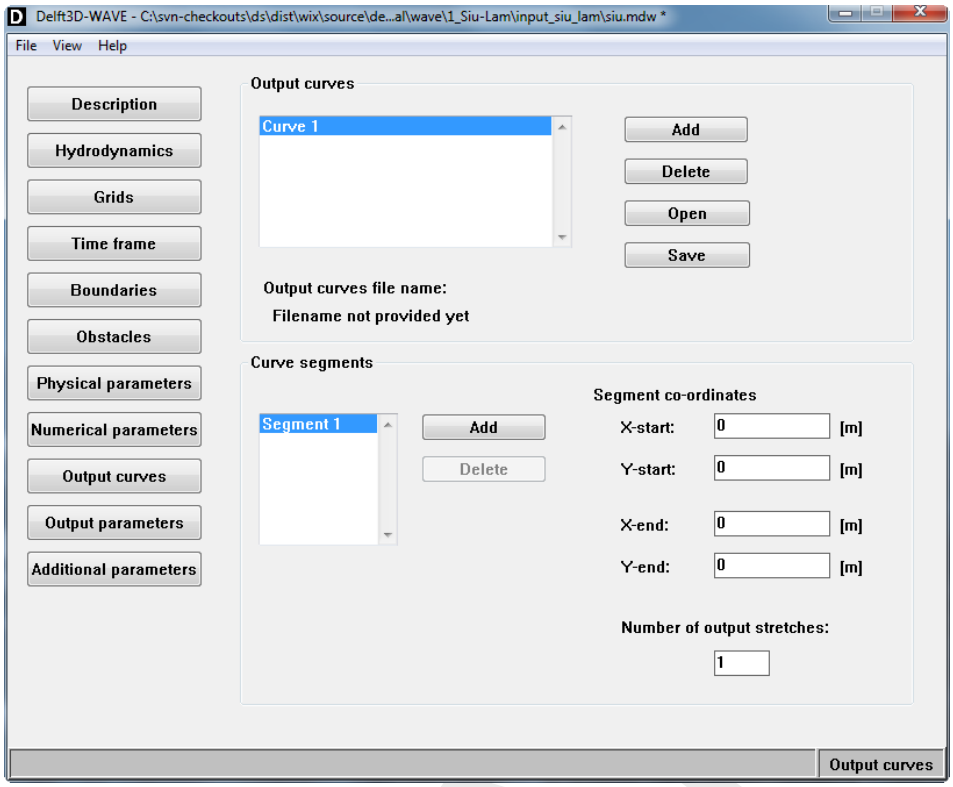

Figure 4.26: *Data Group* Output curves

DIR	mean wave direction (degrees)
DSPR	directional spreading of the waves (in degrees)
DISS	dissipation rate (J m^{-2} s^{-1})
WLEN	mean wave length (in m)
U,V	current velocity (in m/s)

All the data of each output curve is presented in a table and will be saved in only one file, named: <curves.*run-id*>.

4.5.10 Output parameters

Within the Data Group *Output parameters* (see Figure 4.27) you can determine to which grid (i.e. WAVE or FLOW grid) output is written and to which extent the computations should be monitored. The latter option can be used to specify that Delft3D-WAVE should produce intermediate (model) results during a SWAN run (test output) if the program produces unexpected results. Within this data group it is also possible to select output locations for which Delft3D-WAVE produces wave output that is directly obtained from SWAN.

There are three options available to monitor the SWAN computation:

Level of test output (Default: 0)
For values up to 50 test output is made that can be interpreted by you. For values above 50, information for the programmer is produced. For values under 100 the amount is usually reasonable, for values above 200 it can be huge.

Trace subroutine calls (Default: off)
In case an error occurs, the name of the subroutine where the error occurred is written.

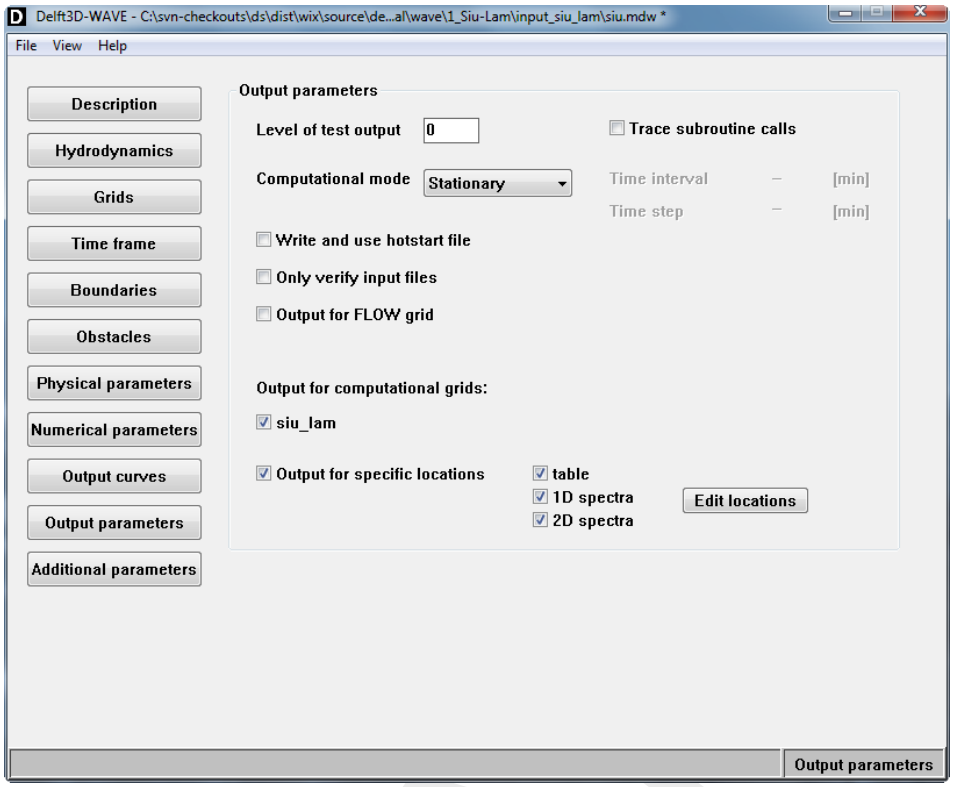

Figure 4.27: *Data Group* Output parameters

Computational mode (Default: Stationary)
Select whether the wave computation is *Stationary* or *Non-stationary*:

> *Stationary* If the *Stationary* option is chosen, and hydrodynamic results from FLOW are used, the *Coupling interval* is displayed as read from the available MDF-file.
> *Non-stationary* In case of the *Non-stationary* option, a *Time interval* (in [min]) for the wave computation should be given. Default value is 0 [min].
> *Non-stationary* In case of the *Non-stationary* option, a *Time step* (in [min]) for the wave computation should be given. Default value is 5 [min].
> *Non-stationary* In case of *Non-stationary* wave computations an alternative numerical scheme is automatically applied. This is because several studies with non-stationary computations have shown that the BSBT numerical scheme performs better in case of non-stationary computations (BSBT: Backward Space Backward Time).

The stationary mode should be used in case of waves with a relatively short residence time in the computational area under consideration, i.e. the travel time of the waves through the region should be small compared to the time scale of the geophysical conditions (wave boundary conditions, wind, tides and storm surge).

Write and use hotstart file (Default: no)
This option can be used to write the entire wave field at the end of a computation to an initialisation file and use this field as initial condition in a subsequent SWAN run. In many cases with a series of wave runs, this option can save significantly amount of computational time. In case of a FLOW-WAVE coupling with a frequent update, the hydrodynamic conditions have not changed a lot since a previous wave computation. Therefore SWAN can use the results of a previous SWAN run as the initial condition for the wave field.

The format of the hotstart file is identical to the format of the files written by the 2D-spectrum output in the pre-defined locations.

Remarks:

- ◇ It is recommended to gradually vary the wave directions in the <wavecon> file. When computing a wave condition using an existing HOT-file, which is generated during a wave computation with a large different wave direction, the use of a HOT-file can lead to unrealistic wave fields. Check the wave results carefully.
- ◇ When applying only one wave condition (e.g. during a flow-wave coupling) it can be wise to increase the required accuracy (in % of wet points) initially. The subsequent wave computations may be completed faster in this way, although the first wave computation will probably need more computational time.

Only verify input files (Default: no)
During pre-processing SWAN checks the input data. Depending on the severity of the errors encountered during this pre-processing, SWAN does not start a computation. You can influence the error level above which SWAN will not start computations. The error level is coded as follows:

- ◇ Warnings
- ◇ Errors (possibly automatically repaired or repairable by SWAN)
- ◇ Severe Errors

Delft3D-WAVE offers two options to save the results of the calculation: on the communication file (if available) and on an output file.

- ◇ *Output for FLOW grid* (Default: off)
  Click in the check box to turn this option on or off. If you select *Output for FLOW grid*, a communication file is available and will be updated. The FLOW model (and other modules) can read and use the wave data directly, since the information is automatically converted to the curvilinear grid definition by the wave module. In section 5.3.2 a description of the output parameters on the communication file is given.
  A curvilinear grid file (FLOW grid) is required to enable this conversion. In case hydrodynamic results from a FLOW simulation are used, the flow input file has been selected. The grid definition is read from this file. If no hydrodynamic results are used, a *Select grid file* button is displayed and a grid file can be selected. If a grid file is selected, still a communication file is needed. The WAVE simulation will expect that the communication file <com-*name*> is available. The communication file can be generated by running a stand-alone FLOW simulation or a online FLOW/WAVE simulation.
- ◇ *Output for computational grids* (Default: off)
  If this option is chosen, detailed output is generated on one or more computational grids. This output is written to a NEFIS file with basename WAVM (waves map file). In section 5.3.2 a description of the output parameters on the <wavm-∗.dat> file is given.
- ◇ *Output for specific locations*
  For the locations to define you can have three types of output: *Table*, *1D spectra* or *2D spectra*.
  Output is generated at user-specified locations; click on *Edit locations* to define the locations manually or by using an input file.
  The parameters written in the *Table* file are:

XP, YP	co-ordinates of output location (with respect to the problem co-ordinates)
DEPT	water depth [m]
HSIG	significant wave height [m]

DRAFT

DIR	mean wave direction [°]
Tpeak	peak wave period [s]
TM01	mean wave period (T_{m01}) [s]
DSPR	directional spreading of the waves [°]
UBOT	root-mean-square value of the maximum of the orbital motion near the bottom [m/s]
XWindv, YWindv	wind components [m/s]
Xvel, Yvel	current velocity components [m/s]

The parameters written in the *1D spectra* file are:

- absolute frequencies [Hz]
- energy densities [J m^{-2} Hz^{-1}]
- average nautical direction [degrees]
- directional spreading [degrees]

The parameters written in the *2D spectra* file are:

- absolute frequencies [Hz]
- spectral nautical directions [degrees]
- energy densities [J m^{-2} Hz^{-1} deg^{-1}]

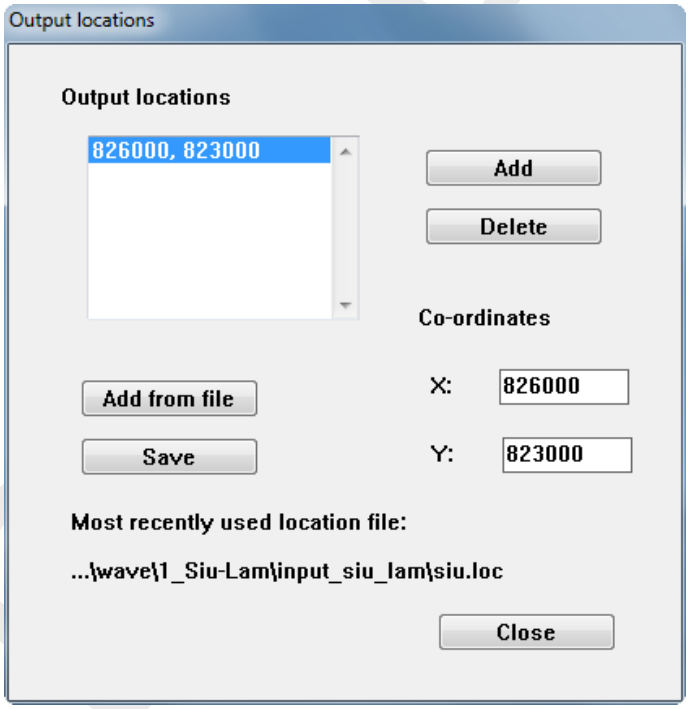

Figure 4.28: *Data Group* Output parameters: Output locations

If *Add from file* is selected, then you should specify this filename. The format of the <∗.loc> file should be:

$$\begin{matrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{matrix}$$

You can also specify manually the x and y co-ordinates by means of the edit boxes.

Remarks:

- ◇ The Table output for specific locations is stored in files <*run-id*nit0j>.tab in case of multiple grids and multiple time points. For the overall computational grid $i = 1$, for the first nested grid $i = 2$, etc. For the first time point $j = 1$, for the second $j = 2$, etc.
- ◇ The 1D spectra output for specific locations is stored in files <*run-id*nit0j.sp1>.
- ◇ Similar for the 2D spectra output in <*run-id*nit0j.sp2> files.
- ◇ In case of only one grid and multiple time points the files are <*run-id*t0j.tab>, <*run-id*t0j.sp1> and <*run-id*t0j.sp2>.
- ◇ In case of multiple grids and only one time points the files are <*run-id*ni.tab>, <*run-id*ni.sp1> and <*run-id*ni.sp2>.
- ◇ In case of only one grid and only one time points the files are <*run-id*.tab>, <*run-id*.sp1> and <*run-id*.sp2>.

4.5.11 Additional parameters

In each datagroup of the mdw-file you can add additional keywords and it's value, this option is used for keywords which are not (yet) supported by the WAVE-GUI. This type of keywords is used for (beta-)testing of new developments on the WAVE-module, the layout of the datagroup is shown in Figure 4.29

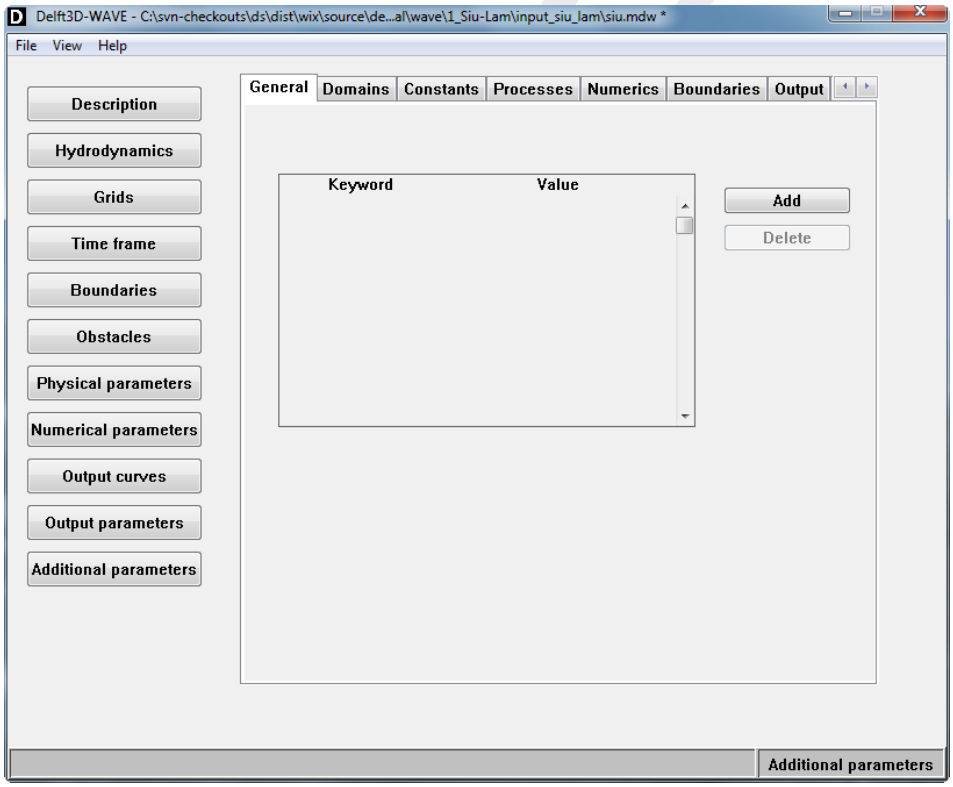

Figure 4.29: *Data Group* Additional parameters

4.6 Visualisation area window

The View menu in the main window allows you to open a visualisation screen. The visualisation screen is built up out of two parts (see Figure 4.30):

- ◇ Pull down menus at the top of the screen.
- ◇ A Visualisation area in the middle.

By opening the pull down menus you are able to open various types of files, to zoom in or out

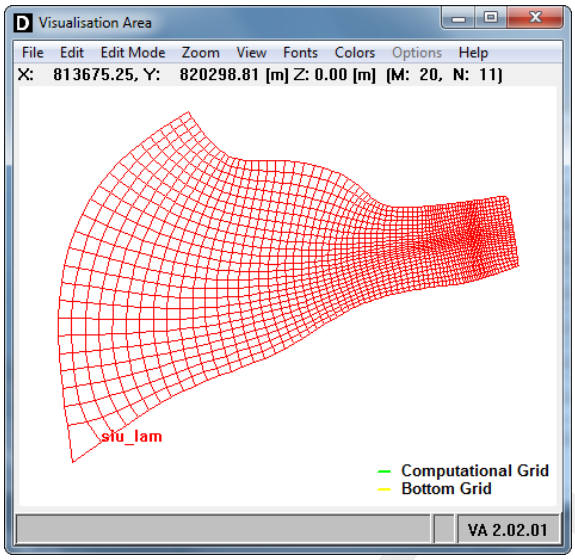
Visualisation Area
File Edit Edit Mode Zoom View Fonts Colors Options Help
X: 813675.25, Y: 820298.81 [m] Z: 0.00 [m] (M: 20, N: 11)
stu_lam
Computational Grid
Bottom Grid
VA 2.02.01

Figure 4.30: *Canvas with Visualisation Area of the wave module*

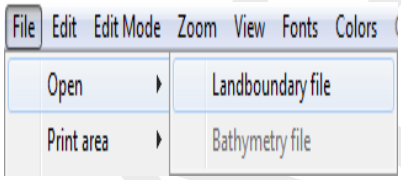

Figure 4.31: File - Open *menu options*

and to set various view options.

In the **Visualisation Area** window all computational grids defined in the Data Group *Grids* are displayed. The grid you are working on in the Data Group *Grids* is highlighted in red. The legend concerning these grids is displayed in the lower right corner of the visualisation area.

Clicking *File* - *Open* enables you to load files and to display additional features, see Figure 4.31.

- ◇ Landboundary file.
- ◇ Bathymetry file (not implemented).

The features could be helpful to locate and position the computational and bottom grids.

Clicking *File* - *Print area* enables you to make a simple screen dump of the Visualisation Area, see Figure 4.32.

With these options the page set-up (i.e. paper size, orientation and scale) can be specified and the print can be made.

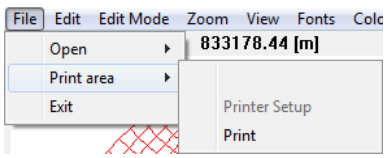

Figure 4.32: File - Print area *menu options*

To leave the **Visualisation Area** window, select *Exit*. The loaded files described above remain loaded for a next visualisation.

Clicking on the options *Edit* and *Edit Mode* will not have any effect since these are de-activated for Delft3D-WAVE.

Clicking on *Zoom* enables you to zoom in or out on the displayed map. If *Zoom Box* is selected you have to use the mouse to drag a box. *Zoom Reset* restores the original zoom level.

When selecting the option *Help*, the version number of the Visualisation Area is given.

4.7 Help function

In the *Help* menu in the main window you can find the *About* option. The *About* menu gives information about the version of the Graphical User Interface. For detailed information about the physics, numeric and commands of SWAN, reference is made to the SWAN-manual (SWAN, 2000).

DRAFT

5 Running and post-processing

5.1 Running

5.1.1 Standalone

Starting point for this section is either you have just finished defining all input parameters of a wave scenario using the WAVE-GUI and you have saved the input data in an mdw-file, or you have available some earlier defined mdw-file.

If you want to use the hydrodynamic results from a finished FLOW simulation, the communication file should be available. Also the following restrictions hold:

- ◇ When using FLOW output, only one com-file can be used; FLOW DomainDecomposition output can not be used.
- ◇ The name of the mdw-file must correspond with the name of the com-file.

See also section section A.1.2.

The **Waves (standalone)** selection window is shown in Figure 5.1.

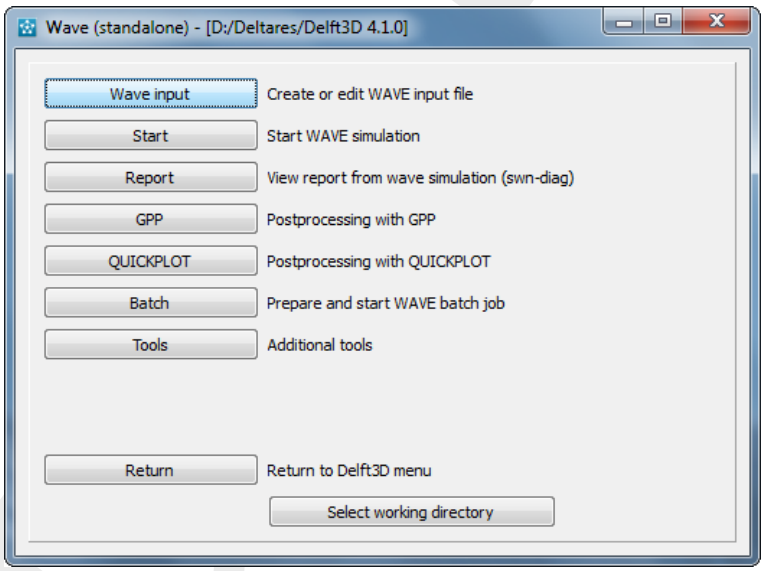

Figure 5.1: Waves (standalone) *selection window for executing a scenario*

5.1.2 Online with FLOW

Starting point for this section is you have prepared both a FLOW and a WAVE scenario. The WAVE scenario can be prepared from the **Waves** selection window, see Figure 5.1, or from the **Hydrodynamics** selection window, see Figure 5.2. In both windows, select *Wave input*.

Restriction:

- ◇ For a FLOW with Online WAVE simulation, both input files must have the same runid.

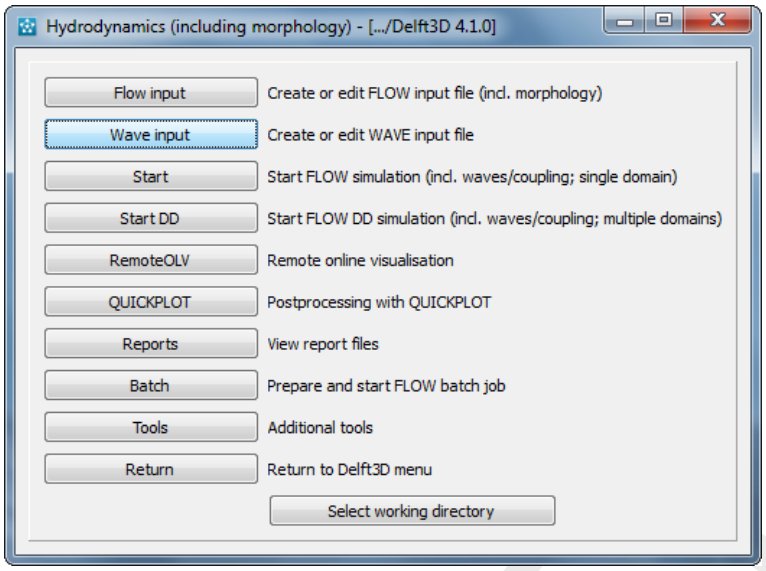

***Figure 5.2: Hydrodynamics** selection window to execute a FLOW-WAVE simulation*

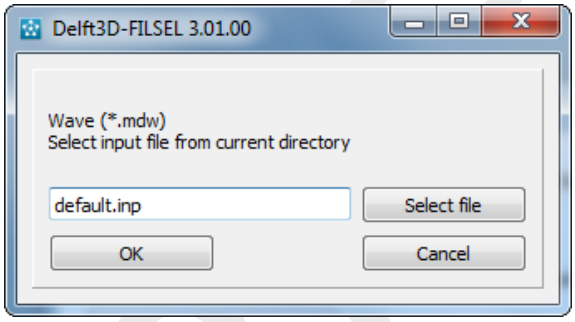

***Figure 5.3:** Select scenario to be executed*

5.1.3 Executing a scenario

After you have prepared the (WAVE and/or FLOW) scenario(s), you can either execute the scenario(s) in foreground or in background. On a Windows based machine (there is not much difference between the two options, but on Linux based platforms there is a large difference. In foreground the status of the simulation and possible messages are displayed in the active window, whereas in background all messages are written to a file and you can continue working in the current window.

- ◇ Select *Start* in Figure 5.1 to carry out a wave (standalone) computation.
- ◇ Select *Start* in Figure 5.2 to carry out a FLOW with Online WAVE simulation.

After this selection, a new window is displayed in which you can select the scenario to be used (see Figure 5.3).

- ◇ Apply *Select file* to navigate through the working directory and select the required:

  <∗.mdw> file for a Wave (standalone) simulation
  <∗.mdf> file for a Flow–Online-Wave simulation

- ◇ Confirm by *OK* and your (Flow-)Wave computation will be carried out.

After the simulation is finished you are strongly advised to inspect at least some of the report

files generated during the simulation to check if all went according to plan. To see the report file of the computation <swn-diag.*>:

◇ Select *Report* in the selection window, see Figure 5.1 or Figure 5.2.

Information on the SWAN computation is found in the <swn-diag.*>.

In case an error is encountered, you should inspect the <*-diag.*> files in your working directory for more information. In most cases you will find a reference to the type of data in which the error was encountered. To correct the error you should:

◇ Close the window in which the simulation was carried out.
◇ Select the *Wave input* option in the Delft3D-MENU.
◇ Open the mdw-file.
◇ Correct the error and carry out the procedure as described in this section until no errors are reported.

Remark:

◇ The number of warnings needs not to be zero for a successful simulation. Still, you are advised always to inspect the warnings and decide for yourself if they are harmless. In cases of doubt, correct the input to resolve the warning.

5.1.4 Files and file sizes

For estimating the required disk space the following files are important:

◇ Waves map file (wavm-file)
◇ Communication file (com-file; only if output is generated on a flow grid file)

Waves map file
The size of the map file is largely determined by the size of the model, i.e. the number of grid points in the computational grid (MXR and MYR). A first rough estimate for the file size of a map file (in bytes) for a computation is: mxr×myr×20.

Communication file
The size of the communication or com-file (e.g. for the other Delft3D modules, such as the FLOW module) from the hydrodynamic simulation is determined by:

◇ The number of grid cells in horizontal and vertical direction: C1.
◇ The number of quantities stored in the simplest simulation: C2.
◇ The number of time steps, for which the communication file is written: C4.

As a first approximation you can take C2 = 15.

For instance, a com-file size of 20.0 Mbytes should be expected for a model containing 50 by 50 points by 5 layers, simulated with density driven currents and simulation results stored for a period of 12 hrs. 30 min. and the file is written with an interval of 15 minutes.

Remark:

◇ The sizes given here are indicative and the figures may not be linearly extrapolated to determine the exact sizes when the number of grid points is enlarged, as these files contain certain types of data which are not dependent either from the intervals or the number of grid points.

DRAFT

5.1.5 Command-line arguments

The following command-line arguments are available to run the computational program <wave.exe>:

```
wave.exe <mdw-file> [mode]
```

<mdw-file>:	Name of the (input) mdw-file
[mode]:	0 Run stand-alone
	1 Run in combination with Delft3D-FLOW
	2 Run in combination with Delft3D-FLOW
	Water and Mud interaction
	default mode = 0

5.2 Frequently asked questions

This chapter aims to help you with common questions that may arise while using Delft3D-WAVE.

1 **Question**
A Delft3D-WAVE run uses the entire CPU of a multicore machine. Can the number of cores being used be forced?
Answer
The parallel version of SWAN is used by default by Delft3D since version 3.28.10. By default, SWAN uses all the cores on the machine. SWAN can be forced to use a specified number of cores, for example 1, by adding the following line to the file <w32/lib/swan.bat> (for Windows):
`set OMP_NUM_THREADS=1`
This line should already be there (line 8), commented out by the tekst "rem " in front of it. The line will be activated by removing the "rem " part.
On Linux, the following line must be added to the file <intel/wave/bin/swan.sh>:
`export OMP_NUM_THREADS=1`
This line should already be there (line 56), commented out by the tekst " # " in front of it. The line will be activated by removing the " # " part.

5.3 Post-processing

5.3.1 Introduction

The post-processors of Delft3D, also known as GPP and Delft3D-QUICKPLOT, offer a comprehensive selection and plotting facility to visualise results. The data used by the GPP model is the data stored in the <wavm-∗.dat> (i.e. the wave map file) and <com-∗.dat> file (communication file; if selected). You can define a single plot or a set of plots and inspect it on screen or make a hardcopy of it on one of the supported hard copy devices. The plots can be processed in an interactive manner or in the background (batch) mode.

In this chapter we only give a very concise description of the post-processors. For a detailed description of their use and functionalities we refer you to the User Manual of GPP and Delft3D-QUICKPLOT.

DRAFT

5.3.2 Model result files of Delft3D-WAVE

Waves map file: <wavm-∗.dat>
If in the Data Group *Output parameters* the option *Output results to computational grid* is selected the <wavm-∗.dat> output file is created. This NEFIS file can be accessed by both post-processors.

The output file presents the results of the calculation on the selected computational grid. The parameters presented below (Table 5.1) are available for post-processing. In Appendix B the definition of the variables is given.

Table 5.1: *Output parameters in <wavm-∗.dat>*

HSIGN	Significant wave height (in m)
DIR	Mean wave direction (direction towards the waves travel in °, measured counter-clockwise from the positive x-axis of the problem co-ordinate system); this direction is the direction normal to the wave crests; note that, if currents are present, it is different from the direction of the energy transport
PDIR	Peak wave direction
PERIOD	Mean wave period of energy density spectrum (in s)
RTP	Relative peak wave period (in s)
DEPTH	Water depth (in m) (not the bottom level!)
FLOW VELOCITY	Current velocity, both the x- and the y-component in the frame co-ordinate system are given (in m/s)
TRANSPORT OF ENERGY	Energy transport vector, both the x- and the y-component with respect to the frame co-ordinate system are given (in W/m)
DSPR	Directional spread of the waves (in °)
DISSIP	Energy dissipation due to bottom friction and wave breaking (in J m^{-2} s^{-1} or N m^{-1} s^{-1})
LEAK	Leakage of energy over the sector boundaries (in J m^{-2} s^{-1})
QB	Fraction of breaking waves (-)
UBOT	The root mean square-value of the maxima of the orbital velocity near the bottom (in m/s).
STEEPW	Mean wave steepness (-)
WLENGTH	Mean wave length (in m)
TPS	Smoothed peak wave period (s)
TM02	Mean absolute zero-crossing period (s)

DRAFT

TMM10	Mean absolute wave period (s)
DHSIGN	Difference in significant wave height during last iteration (m)
DRTM01	Difference in average wave period during last iteration (s)
SETUP	Set-up due to waves (only when activated; in m)
WAVE FORCE	Wave-induced forces (FX, FY in N/m^2)
WIND	Wind velocity (WINDU, WINDV in m/s)

Communication file: <com-∗.dat>
If in the Data Group *Output parameters* the option *Output for FLOW grid* is selected, the <com-∗.dat> output file is updated. This NEFIS file can be accessed by the Delft3D post-processors or can be used as input for a wave-induced flow calculation (Delft3D-FLOW).

Delft3D-WAVE writes the `wavtim` group to the communication file. The `wavtim` group concerns the computed wave parameters, for the times `timwav`, being the times specified in the Data Group *Time frame*.

The output file presents the results of the calculation on the selected flow grid. The parameters presented below (Table 5.2) are available for post-processing. In Appendix B the definition of the variables is given.

Table 5.2: *Output parameters in <com-∗.dat>*

HRMS	Root mean square wave height (in m)
TP	Peak wave period (in s)
DIR	Mean wave direction (direction relative to the flow grid in $^\circ$, measured counter-clockwise); this direction is the direction normal to the wave crests; note that, if currents are present, it is different from the direction of the energy transport
DISS	Wave energy dissipation rate due to bottom friction and wave breaking (in W/m^2 or N m^{-1} s^{-1})
FX, FY	Wave forcing, both the u- and the v-components (in N/m^2)
MX, MY	Wave-induced volume flux, both the u- and the v-component (in m^3/sm)
TPS	Smoothed peak wave period (s)
UBOT	The root mean square-value of the maxima of the orbital velocity near the bottom (in m/s).
WLENGTH	Mean wave length (in m)

DRAFT

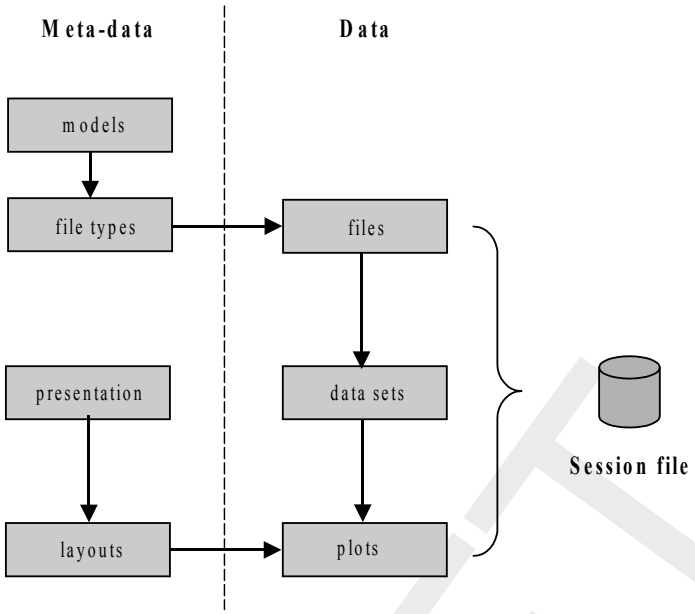

Figure 5.4: *Hierarchy of GPP*

5.3.3 Working with GPP

GPP offers a comprehensive selection and plotting facility to visualise or animate simulation results, to import and visualise other data such as measurements, or to export selected data sets of the results for use in other programs. You can define a single figure or a set of figures and inspect it on screen or make a hardcopy of it on one of the supported hard copy devices. The figures can be processed in an interactive manner or in the background (batch) mode. In this section we only give a very concise description of the post-processor. For a detailed description of its use and functionalities you are referred to the GPP User Manual (GPP, 2013).

Overview

When executing a project with many simulations the amount of data from which a set will be visualised can be enormously large; also the files in which the data are stored can be very large. Therefore it is not optimal to search the original result files over and over again for each parameter or for each new figure. GPP, instead, provides a mechanism to make a selection of the various results and parameters before starting the actual visualisation process and makes a kind of reference list to these sets of data. Next you can define one or more graphs and fill them with data from these data sets. As GPP knows were to find this data, retrieving the data is executed very efficiently. This efficiency is further increased by the option to select all observation points for a certain quantity or to select all time instances at which a quantity is stored in the map file and let you make the final selection when producing the figure.

GPP has access to the communication file and to the result files of all Delft3D modules and in fact to many other programs of Deltares, so you can combine almost any kind of data in a figure.

GPP uses a certain hierarchy in the data and the meta-data, see Figure 5.4.

We distinguish meta-data to specify the definitions of a figure at a high level of abstraction (left part of Figure 5.4) and the actual data (right part of Figure 5.4).

DRAFT

Meta-data:

models	Defines the set of models the results of which can be visualised. You can change this set of models to limit the options presented in GPP menus.
file types	Defines the set of file types that can be used. You can change this set of file types to limit the options presented in GPP menus.
presentations	Defines the set of data-presentation methods, such as contour maps or xy-graphs.
layouts	Defines the set of layouts that can be used in a figure. You specify the general set-up of a figure by defining the appropriate layout, the size of the graph, the plot areas, their position, additional text etc.

Data:

files	The actual files to be used in your plot session.
data sets	The actual data sets selected from the files and to be used in the visualisation.
plots	The actual figures, including the data, which will be presented on your monitor or printed on paper.
session file	An ASCII file containing all information that defines the figures. For the data only the references to the data is stored in the session file, not the data itself.

Launching GPP

To start GPP, select from the Delft3D-MENU *Wave - GPP* and next Figure 5.5 is displayed.

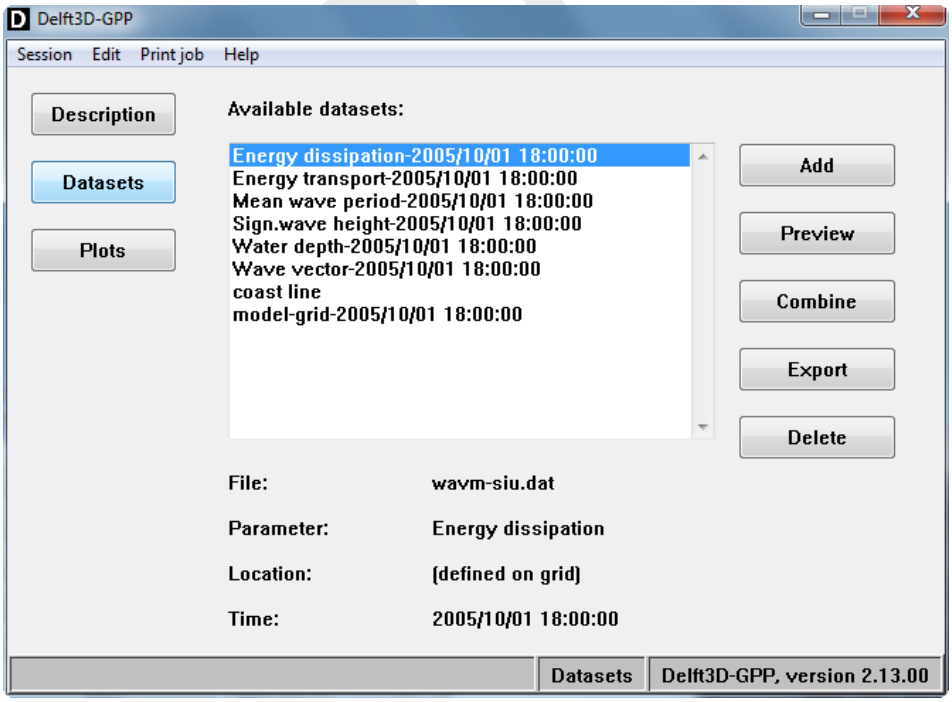

Figure 5.5: *Main window of GPP*

The basic functions are shortly described below; for full details you are referred to the GPP User Manual.

Session	To load an existing session file or to save the settings and selections

	of the current session in a session file for later use.
Description	To give a short description of a session file; this information is used only for reference.
Datasets	List of pre-selected data sets to be used in the current plot session. You can give selected data sets a useful name. At start-up the selections are displayed of the previous plot session in the current directory.
Plots	List of pre-selected plot layouts to be used in the current plot session. At start-up the selections are displayed of the previous plot session in the current directory.
Add	To add a data set or plot layout, depending which function on the left side of the listbox has been selected.
Preview	To preview a selected data set or plot layout from the list displayed in the listbox.
Combine	To combine any of the available (single) data sets to a new data set, such as multiply, divide, take the maximum value etc. and save the new data set under a unique name.
Export	To export the selected single or combined data set to an ASCII file or GIS-file (for single data sets only).
Delete	To delete the selected data set or plot layout.

To add a data set of a specific result file to the *Available data sets* in Figure 5.5:

- ◇ Select *Datasets - Add* in Figure 5.5.
- ◇ Click *Select File* in the **Add dataset** window, Figure 5.6.
- ◇ Select the required data file in the file selection window that is being displayed.

The parameters and locations (or time in case of map-results) available in a selected result file are displayed in the **Add dataset** window, see Figure 5.6.

You can make as many selections from a specific result file or from different results files (to combine results from different computations or models) as you like.

To have a quick view on a data set:

- ◇ Select in Figure 5.5 the required data set and click *Preview*.

The selected data set and a default plot layout will be displayed in the **Plot window**, see Figure 5.7.

Remark:

- ◇ GPP recognises the type of data selected and uses an appropriate default presentation method to display the results.

You are referred to the User Manual of GPP for full details on how to use GPP.

DRAFT

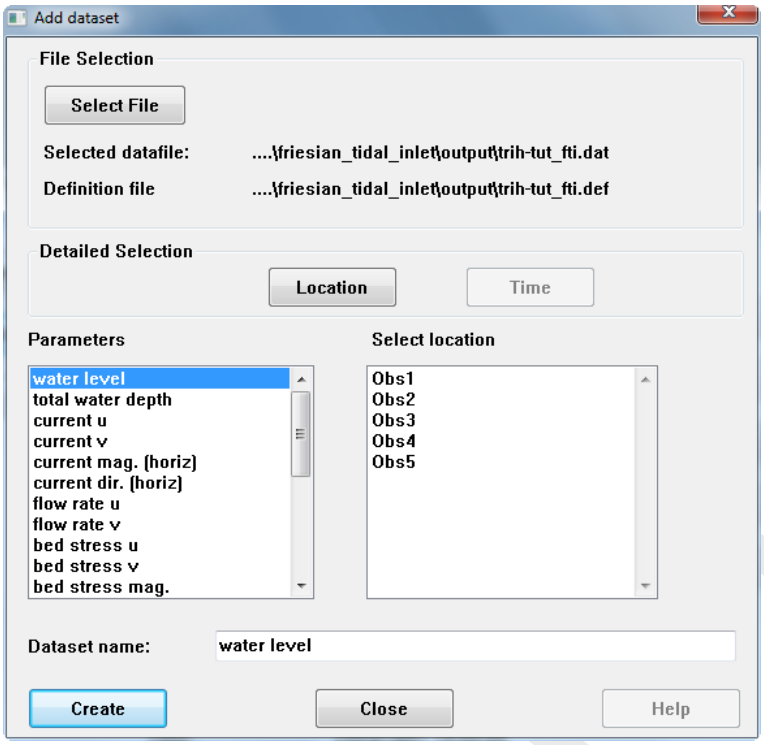

Figure 5.6: *Parameters and locations in the <trih-tut_fti.dat> file*

5.3.4 Working with Delft3D-QUICKPLOT

Basically there are just four or five steps to get your first plots using Delft3D-QUICKPLOT: start the program, select the file, select the data field, select the time and location, and press plot. The following text will show you how to get your first plots of some Delft3D-FLOW map and history files (other files can be processed in exactly the same way).

Starting the program

Delft3D-QUICKPLOT can be started from the Delft3D-MENU by selecting *Utilities - QUICKPLOT*. Alternatively, you can run the program `d3d_qp.exe` from the directory <$D3D_HOME/$ARCH/quickplot/bin/> .

As the program starts, the main program window appears. It will initially look as shown in Figure 5.8. The left part of the window contains the fields for opening and closing files, selecting data sets, time steps and plotting locations, and the buttons for creating the actual plots. The right part of the window (now empty) will contain all options for the selected data set (plot and export options).

Selecting a data file

The first step in creating a plot is opening a data file. This can be accomplished by clicking on the *Open a data file* toolbar button or by selecting *Open File* from the *File* menu.

From the standard file selection window that appears select the data file you want to process. The selection window contains a number of pre-configured filename filters, such as Delft3D output file <∗.dat> and Delft3D grid file <∗.grd>.

Remarks:

- Although the selection interface lists for the Delft3D output files only the data files <∗.dat>, the accompanying definition files <∗.def> are always required for reading

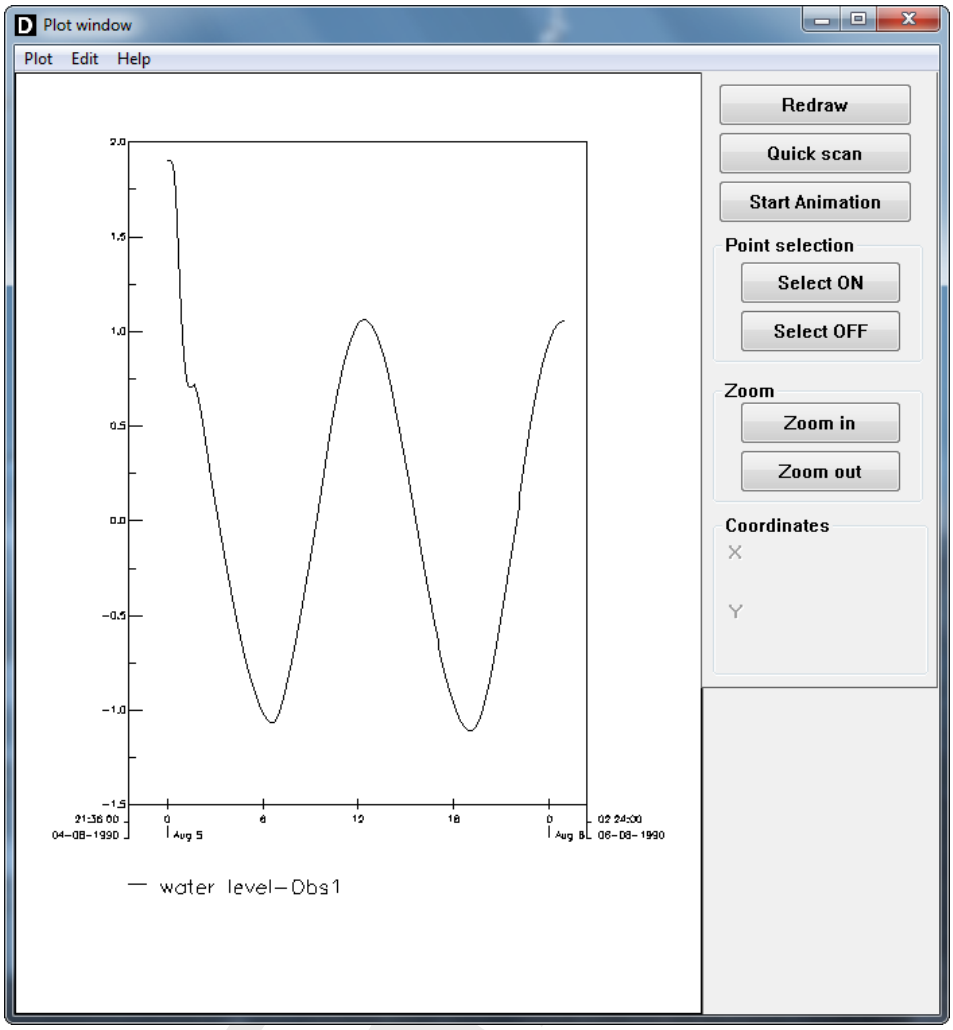

Figure 5.7: *Plot window of GPP*

the data files.

◇ The filename filter does not influence the automatic recognition procedure that follows the selection procedure, so any file may be selected with any filename filter active.

After opening a Delft3D-FLOW map-file, the Delft3D-QUICKPLOT interface will activate a larger part of its interface. It will look as shown in Figure 5.10. The filename is indicated as the active file in the dropdown list just below the *Open a data file* button. Below the filename, the data fields available from the selected file are shown. The *Quick View* button for plotting the result is activated, and some plotting and export options are available from the right part of the window. This basically indicates that you can already create your first plot now, but let us first inspect the other parts of the interface.

Selecting a data field

The next step in creating a plot is selecting the quantity or data field from the file to be plotted. The data fields available from the active file are shown in a dropdown list below the name of the file. Click on the selected field (in the example: 'wave grid') to expand the list and to select another data field as shown in Figure 5.11. The supported file formats and the data fields that may be contained in them are listed in Appendix A of the Delft3D-QUICKPLOT User Manual.

Different quantities allow for different types of plots and, therefore, the lists of plot and export

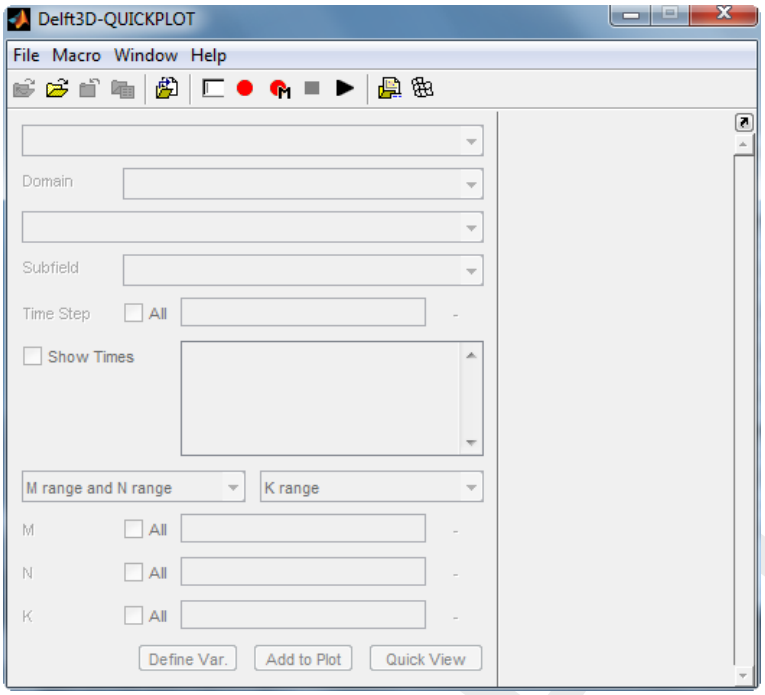

Figure 5.8: *Delft3D-QUICKPLOT main window*

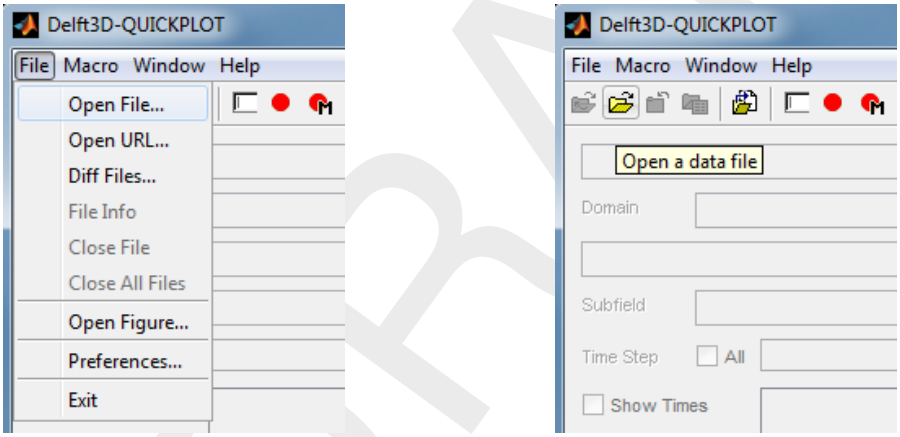

Figure 5.9: *The 'File Open' command can be selected in two ways*

options in the right part of the window will adapt to your selection. Figure 5.12 shows the list of options if the 'hsig wave height' (or any other scalar 2D quantity) is selected. Furthermore, the number of time steps depends on the selected data field; the example file contains 3 time steps for the wave height as indicated by the edit box below the data field listbox.

The domain selection box between the file selection box and the data field selection box is only active when the file may contain multiple domains. Similarly, the sub-field selection box immediately below the data field selection box is only active when the data field contains multiple sub-fields (e.g. the data field 'sediment transport' may have sub-fields for sediment fractions 1, 2, etc.)

Selecting time and location

After the selection of the data file and the data field, you must select which time step and which location to plot. The default setting is to plot the last time step in the file and the whole

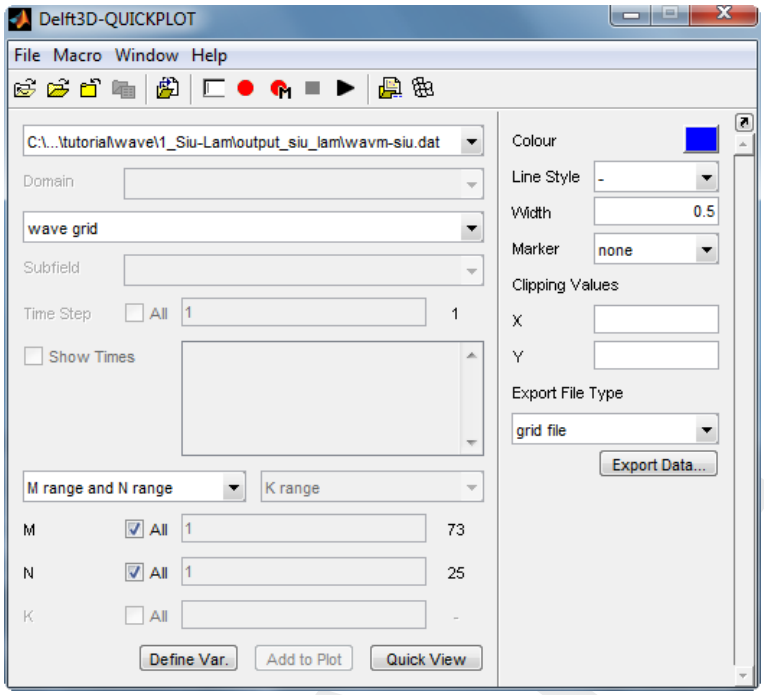

Figure 5.10: *User interface after opening a Delft3D-WAVE map file*

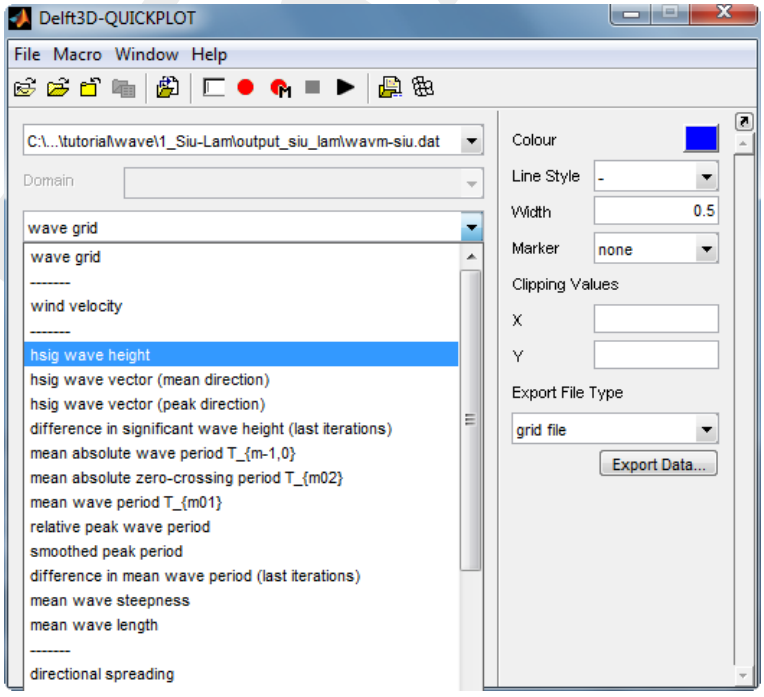

Figure 5.11: *List of data fields in the Delft3D-WAVE map file*

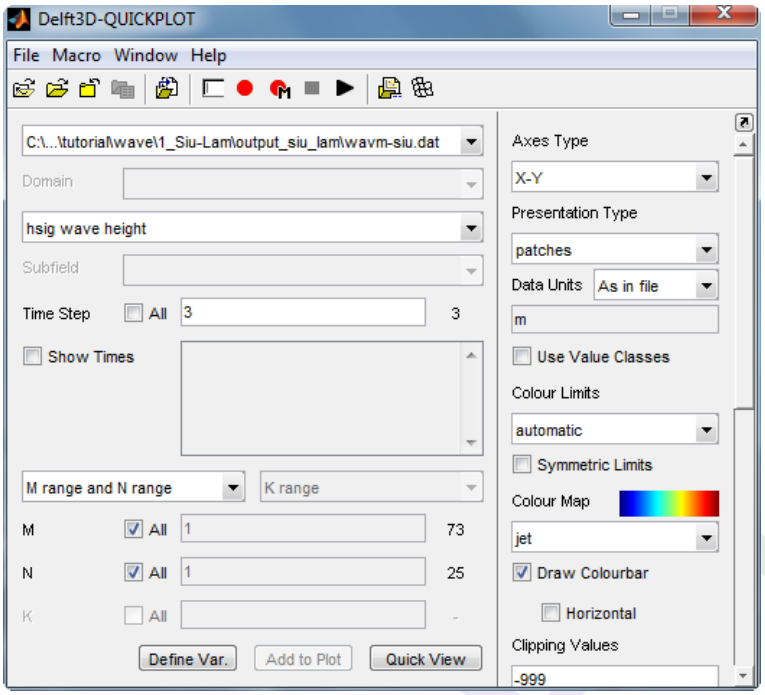

Figure 5.12: *List of plot options is changed after selection of the* hsig wave height *from the dropdown list*

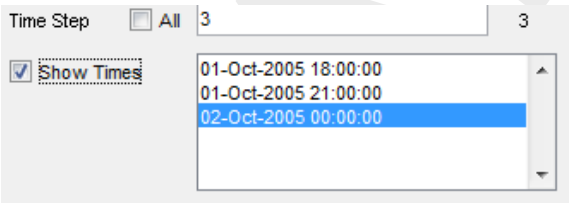

Figure 5.13: *Optional listing of the times associated with the various time steps*

domain. In the case of Figure 5.12, this is indicated by the selection of time step 6 and all M and N indices.

Remark:

- ◇ If you want to see the times associated with the time steps stored in the file, tick off the *Show Times* checkbox (see Figure 5.13). Reading and displaying a large number of times can be very time consuming and you should be careful when opening data files (generally history files) containing a large number of time steps: uncheck the *Show Times* checkbox first.

If instead of a 2D plot of the whole domain, you want a plot of a cross-section along an M grid line uncheck the *All* checkbox associated with *M* and specify the M-value of the desired grid line as shown in Figure 5.14.

Remark:

- ◇ The valid range of grid and time step numbers is indicated to the right of the M/N/K and time step edit boxes, respectively. The indicated range of grid points includes the extra row of points added due to staggering of the variables on the computational grid. Depending on the selected data field, the first and last grid lines may or may not have data defined on it.

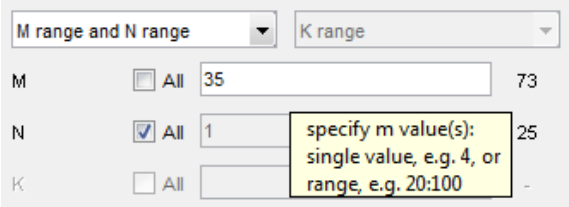

Figure 5.14: *Selection of a cross-section along a grid line in M direction: one M value, all N values*

If you want a time-series plot at any computational point of the grid, select *All* (or multiple) time steps and one M and one N (and optionally one K) index.

Remark:

- The extraction of a time-series from a map-file is carried out by reading for each selected time step the whole domain and selecting only the requested point. This procedure is more flexible yet also slower than selecting history points in the Delft3D input.

Creating a plot

You can now plot the data by pressing the *Quick View* button. Depending on the data field selected, the selected time step and the selected spatial extent, you will get a 2D plot, a cross-sectional plot or a time-series plot. Figure 5.15 shows a result.

Remarks:

- If you have selected multiple time steps and a spatially extended plot domain (i.e. all or multiple M, N or K co-ordinates), the *Quick View* button will have changed into a *Quick Animate* button. Pressing the button will cause the program to animate the selected plot by looping over the selected time steps. The same result can also be obtained by selecting one time step initially and using the *Animation* menu in the plot.
- It is currently not possible to plot data sets on a 3D domain (i.e. all or multiple M, N and K indices selected). Always specify a single M, N or K index for 3D data sets.

If there are multiple time steps and if you have selected only one, or if you have selected only one M, N or K index, the plot will contain an active slider in the lower left corner of the plot. You can select other time steps and other spatial co-ordinates using that slider.

DRAFT

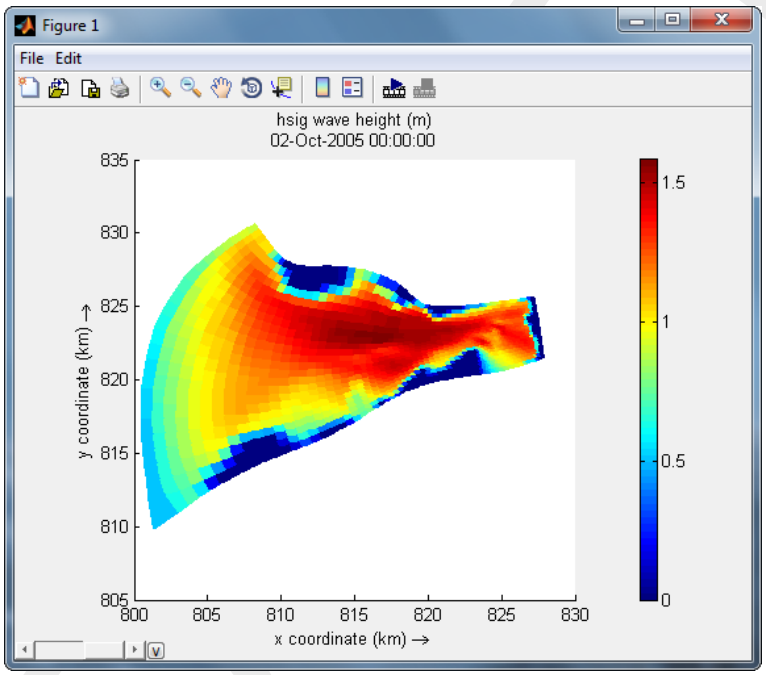

Figure 5.15: *2D Plot of the 'hsig wave height*

6 Tutorials

6.1 Introduction

In these tutorials we will guide you through the process of creating a simple example of a wave computation with SWAN. All the information for a wave computation, also called a scenario, is stored in an input file, also known as Master Definition Wave file (MDW-file). However, before starting this input definition process we want to explain in short the basics of a model definition, the structure of an MDW-file and the basic steps you are supposed to execute.

To execute a wave computation for a specific area we need various kinds of information, such as the extent of the model area, i.e. the boundary at which the incident waves are prescribed, the wind, the bathymetry, geometrical details of the area such as obstacles and a selection of the results that need to be stored for later inspection. Finally, a numerical grid must be defined onto which all location related parameters are defined. So, the basic steps that precede the definition of an input file can be summarised as:

- ◇ Selection of the extent of the area to be modelled.
- ◇ Definition of location and type of wave boundary.
- ◇ Generation of the bathymetry defined on the grid.
- ◇ Definition of many different options, such as wind speed and direction, water level field, current field, number and type of obstacles, etc.

Some of these activities (such as the generation of the bathymetry) must be done before starting the WAVE Graphical User Interface (GUI). In most cases they result in one or more files that are to be located in a project directory to be defined when starting the project. The project directory is also referred to as the working directory. The first two steps are based on experience in solving similar problems and on engineering judgement, no tools are available to support these steps others than (GIS-based) maps and (digitised) charts.

The data of the land boundary, the bathymetry (and the numerical flow grid if present) are stored in separate, so-called attribute files. In the MDW-file only a reference is made to these files instead of including all data in the MDW-file itself. The advantage of using attribute files is that the data can be used in many scenarios but it is stored only once on the system disks. However, the user himself must keep some administration on the use of the same attribute files in different scenarios.

For these tutorials the files, which are created outside the GUI, are provided.

6.2 Siu-Lam wave model (1 grid; 3 wave runs)

6.2.1 Introduction

In this tutorial we provide an existing MDW-file with attribute files for a specific example called ‘Siu-Lam’. The area modelled concerns an estuary called *Siu-Lam* near Hong Kong. We use this basic example to guide you through most of the input definition part of a wave simulation. It is noted that the wind, wave and other parameters that are used do not represent realistic conditions for that area. Therefore the presented results have no practical use.

The input data is located on the directory

<. . . /tutorials/wave/1_Siu-Lam/input_siu_lam>

and need to be copied first to the directory

DRAFT

<. . . /tutorials/wave/1_Siu-Lam/>.

The files used in the case of *Siu-Lam* are:

- ◇ <hongkong.ldb>: land boundary file
- ◇ <siu_lam.grd>: grid file
- ◇ <siu_lam.dep>: bathymetry file

The land boundaries <∗.ldb> and the (Delft3D-FLOW) grid <∗.grd> and bathymetry <∗.dep> files can be helpful to design the computational grids for the wave model.

In this tutorial we will use area averaged values for the water level in stead of hydrodynamic results of a Delft3D-FLOW calculation.

6.2.2 WAVE Graphical User Interface

To start the WAVE Graphical User Interface (GUI), execute the following commands (see chapter 3 for details):

- ➢ Click the Delft3D-MENU icon on the desktop (PC) or execute the command `delft3d-menu` on the command line (Linux).
- ➢ Select the item *Wave*.
- ➢ Change to the working directory; in this tutorial <. . . \tutorial\wave\1_Siu-lam>.
- ➢ Select *Wave input* in the **Waves (standalone)** selection window to start the WAVE-GUI.

The start-up window of the WAVE-GUI will be displayed (see Figure 6.1).

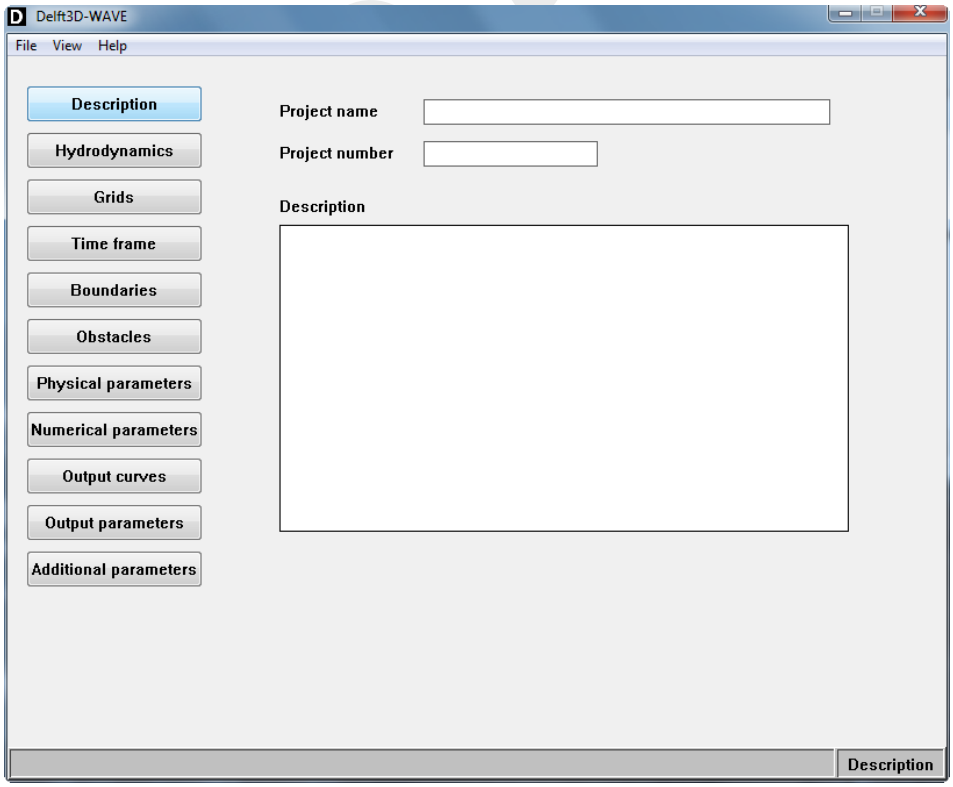

Figure 6.1: *Starting window of the WAVE Graphical User Interface*

Now you are in the main window of the WAVE-GUI. We have not selected an existing MDW-

file, because by doing so we would automatically have loaded all the attribute files referred to in the MDW-file. Instead, you are going to define yourself all the input that is part of the tutorial scenario Siu-Lam.

You are now ready to start defining your own scenario.

6.2.3 Saving input data

Initially, this tutorial may be somewhat tedious to work on. Rather than going on until the end, you may want to stop somewhere halfway the exercise. To prevent that you have to enter all the data again when restarting the exercise, you should save the data you have entered. In this case:

➢ Go to the *File* item in the menu bar of the WAVE-GUI window.
➢ Click *Save As.*

Go to the working directory and save the MDW-file under a new or under the same name (overwrite).

Remark:

◇ Upon saving the data, the GUI checks its integrity and will show you a message window if needed. Adjust your data untill there are no warnings/errors anymore before saving the MDW-file.

6.2.4 Data groups

The Delft3D-WAVE input is divided into several data groups. By selecting a button you get access to a data group. Each of these data groups will be described in the following sections.

6.2.5 Description

In the Data Group *Description* you can identify this MDW-file by giving a comprehensive description of the project, the application domain and the specific selections to be made in this scenario. The description is only used for identification and has no influence on the simulation itself.

➢ Type the description as displayed in Figure 6.2.

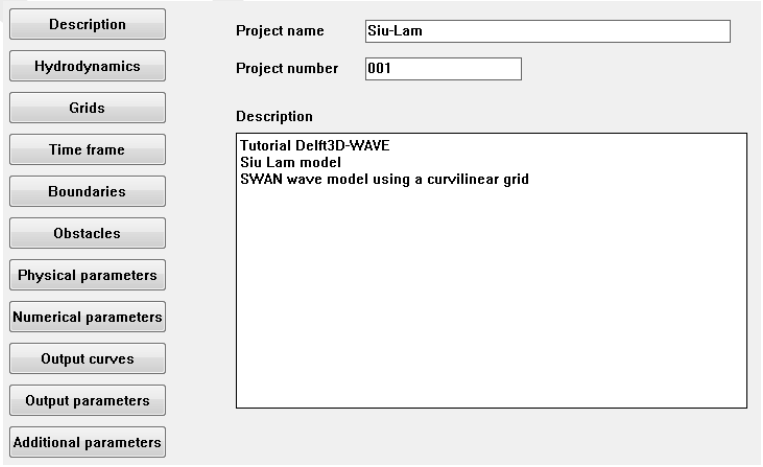

Figure 6.2: *Data Group:* Description *and sub-window*

6.2.6 Hydrodynamics

With Delft3D-WAVE you can run a wave computation that uses results from the FLOW module but also a standalone wave computation. This tutorial will not use FLOW results, in stead area averaged hydrodynamic values will be specified in Data Group *Time frame*.

6.2.7 Grids

In the Data Group *Grids* you define the computational grid(s) with the corresponding bathymetry file(s) (see Figure 6.3). In addition, the spectral grid on which SWAN performs the computation has to be specified per computational grid. When importing more than one grid, the nesting relations should be specified.

The grids can be defined in a common Cartesian co-ordinate system or in a spherical co-ordinate system, described in chapter 7. The choice of co-ordinate system should already be made when the grid is generated using RGFGRID.

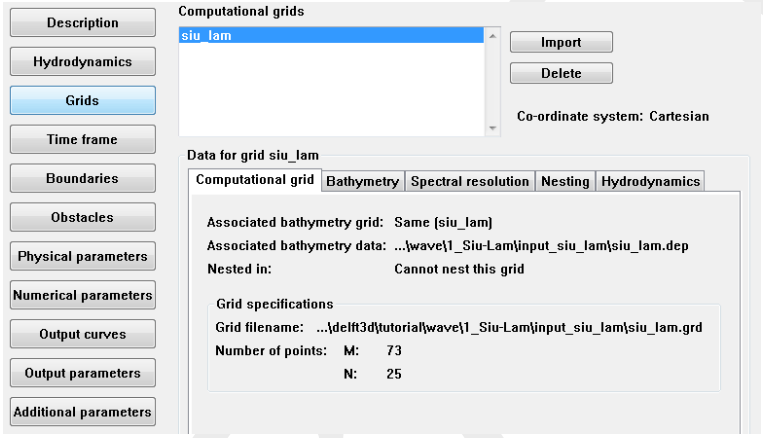

Figure 6.3: *Data Group* Grids

A computational grid is a grid on which SWAN solves the wave action balance equation. Within Delft3D-WAVE, SWAN wave computations can only be made on a curvilinear grid (which can still be rectangular, but created with RGFGRID). Each computational grid has its own corresponding bathymetry file <∗.dep>, created with QUICKIN. This file should be selected under the tab **Bathymetry**.

6.2.7.1 Computational grid

In Delft3D-WAVE you can specify several grids in one run; you have to point out which grid is nested in which. The idea of nesting is to have a coarse grid for a large area and one or more finer grids for smaller areas. The coarse grid computation is executed first and the finer grid computations use these results to determine their boundary conditions. Nesting can be repeated on ever decreasing scales. Additional information on this topic is given in section 4.5.3.4 (Nesting) and section 7.2.2 (Choice of grids and boundary conditions).

- Select the *Grids* Data Group (Figure 6.3).
- Select *Import* to load a computational grid. Select from the browse screen the desired file: <siu_lam.grd>. Click *Open* to confirm the operation.

The steps above can be repeated when more grids need to be imported.

The grids can be displayed in the **Visualisation Area** window by selecting *View* → *Visualisation Area* from the menubar.

➢ Select *File* → *Open* → *Landboundary file* from the pull down menu in the **Visualisation Area** to open the land boundary. Select from the browse screen the desired file: <hongkong.ldb>.
➢ Click *Open* to confirm the operation. The land boundary will be displayed in the **Visualisation Area** window.
➢ Select *Zoom* → *Zoom Box* from the pull down menu in the **Visualisation Area**. Push and hold the mouse button, drag a box and release the button to zoom in on the concerning area (see Figure 6.4).

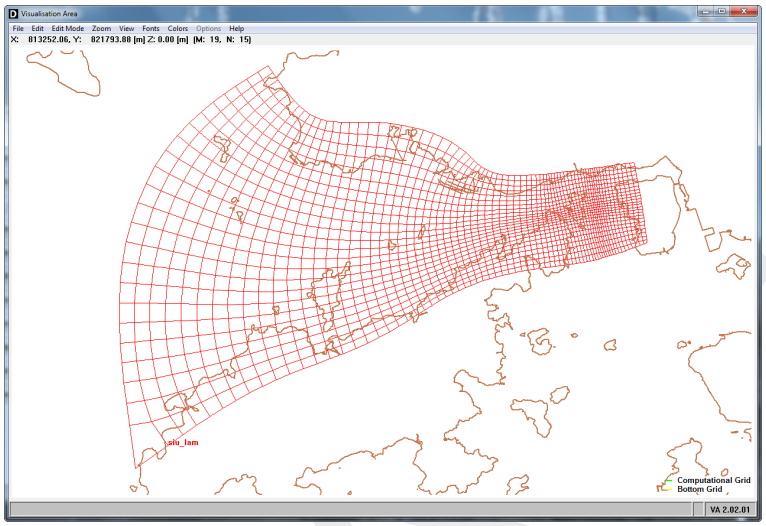
Visualisation Area
File Edit Edit Mode Zoom View Fonts Colors Options Help
X: 813252.06, Y: 821793.88 [m] Z: 0.00 [m] (M: 19, N: 15)
siu_lam
Computational Grid
Bottom Grid
VA 2.02.01

***Figure 6.4: Visualisation Area** window*

6.2.7.2 Bathymetry

The bathymetry data can be defined on the corresponding computational grid, but the bathymetry can also be provided on another grid; this grid must be rectangular. This grid should again be generated using RGFGRID. The bathymetry data should then be provided based on this rectangular grid using QUICKIN.

➢ From the tab **Bathymetry** (see Figure 6.5) click *Select bathymetry data* to open the corresponding bathymetry.
➢ Select from the browse screen the desired file: <siu_lam.dep>. Click *Open* to confirm the operation.

6.2.7.3 Spectral resolution

The computational grids have now been defined for SWAN. In addition to the computational grids in geographical space, SWAN also calculates wave propagation in the spectral space (see section 7.2.2). To that end, for each geographical grid the spectral grid has to be specified using the **Spectral resolution** tab.

➢ Click the tab **Spectral resolution** (see Figure 6.6) to edit the spectral grid for each computational grid (i.e. coarse and nested grids).

In the canvas *Directional space* you can define the range and the resolution in directional space for SWAN. In the present example the *Circle* option is considered (this means the full circle of 360° is taken into account). Edit the box *Number of directions* to specify the number of spectral directions.

➢ Enter the value “36” ($\Delta\theta$ = 360°/36).

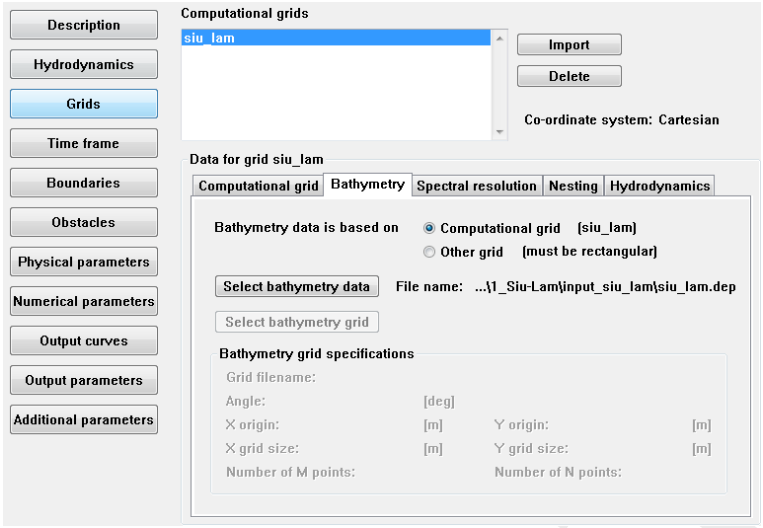

Figure 6.5: *Sub-data Group* Bathymetry

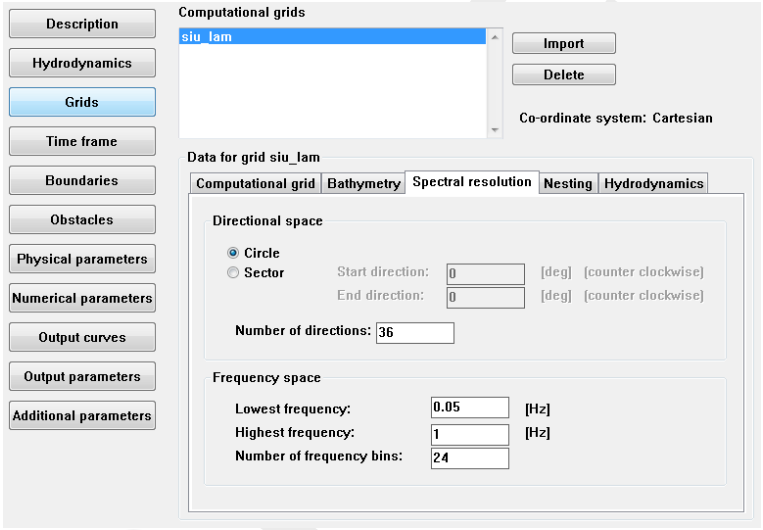

Figure 6.6: *Sub-data Group* Spectral resolution

In the canvas **Frequency space** you can define the resolution and the range in frequency space. The *Numbers of frequency bins* are the numbers of meshes in the frequency-space (one less than the number of grid points in frequency space). This defines the grid resolution in frequency space between the *Lowest frequency* and the *Highest frequency*. This resolution is not constant since the frequencies are distributed logarithmic (see section 7.2.2).

◇ For the current computation you can leave the values as default:

Lowest frequency: 0.05 Hz
Highest frequency: 1 Hz
Numbers of frequency bins: 24

◇ Select (if available) the other computational grids and specify the spectral space resolutions for all grids as for the first grid.

Remarks:

◇ If you want to consider only wave directions in a limited directional sector, the option *Sector* may be chosen. The range in Cartesian degrees of this directional sector is

specified giving the *Start direction* and the *End direction*.
- ◇ SWAN has the option to perform computations on a nested grid. In such cases, the spectral resolution of the nested grid does not need to be equal to the spectral grid of the coarse grid.

6.2.7.4 Nesting

When you want to make nested runs, you first have to import all considered grids. In *Grids* → *Nesting* you must prescribe in which grid the selected grid should be nested. An example of a nested wave model can be found in section 6.3.

Remarks:
- ◇ The first grid cannot be nested in another one. For this grid, boundary conditions must be specified in the Data Group *Boundaries*.
- ◇ A grid cannot be nested into itself.
- ◇ If land points remain dry during the computations, then these points will be ignored for the SWAN computation.

6.2.7.5 Hydrodynamics

For this tutorial the default settings of *Hydrodynamics* will be used.

6.2.8 Time frame

In the Data Group *Time frame* a number of times are specified at which wave computations must be carried out. If the hydrodynamics results of a FLOW simulation are used (see Data Group *Hydrodynamics*) then the time points can be selected at which these results are available. In this tutorial we will specify time steps and use a default (uniform) water level and velocity.

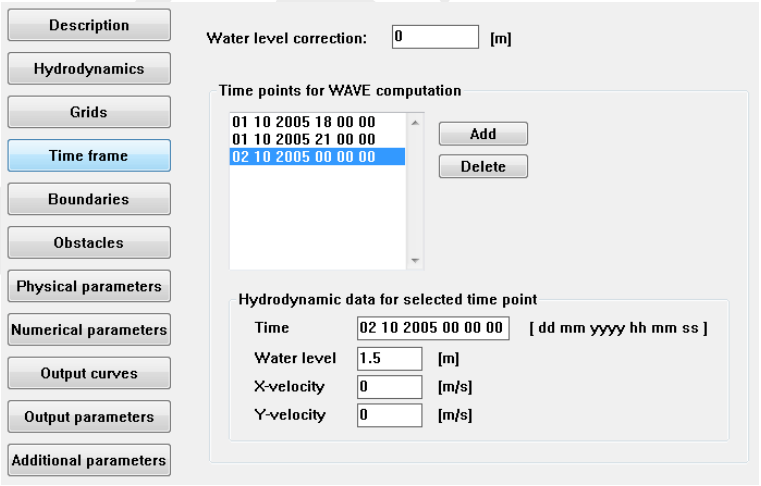

Figure 6.7: *Data Group* Time frame.

- ➢ Select *Time frame* to enter the Data Group (see Figure 6.7). No *Water level correction* is applied (0 m).
- ➢ Press *Add* to define a time point for a wave simulation. Specify in the *Time* input field “01 10 2005 18 00 00”.
- ➢ For this time point enter for the *Water level* “-1.0” and for the *velocities* “0”.
- ➢ Press *Add* to define a time point for a wave simulation. Specify in the *Time* input field “01 10 2005 21 00 00”.
- ➢ For this time point enter for the *Water level* “0.0” and for the *velocities* “0”.

- Press *Add* to define a time point for a wave simulation. Specify in the *Time* input field “02 10 2005 00 00 00”.
- For this time point enter for the *Water level* “1.5” and for the *velocities* “0”.

You can click *Delete* to remove a selected time point from the *Time points for WAVE computation* list.

6.2.9 Boundaries

In the Data Group *Boundaries* the incident wave conditions at the boundary of the first computational grid are prescribed (see Figure 6.8). All other computational grids (i.e. the nested grids) obtain their boundary information from other grids.

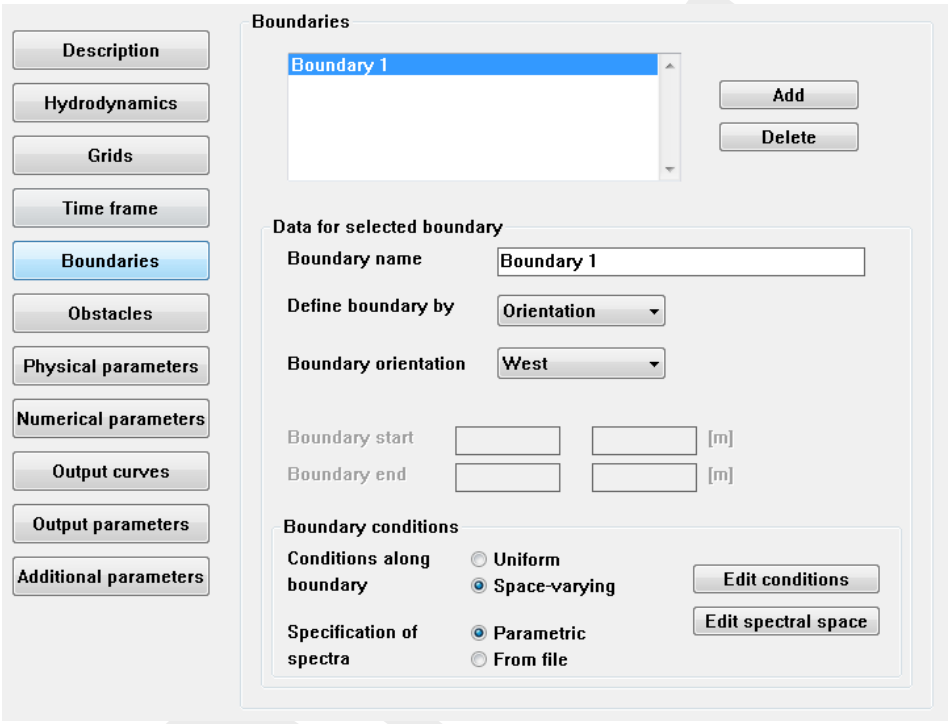

Figure 6.8: *Data Group:* Boundaries

In the SWAN computations, wave boundary conditions may be specified at each side of the computational grid (i.e. maximum of 4 sides). The number of sides at which boundary conditions are provided is zero by default.

The up-wave boundary in the Siu-Lam example is the boundary along the west side of the first computational grid. The wave conditions can vary along this up-wave boundary.

The boundary conditions in SWAN can be defined by specifying the integral wave parameters or can be read from an external file (i.e. results of other model runs or field observations).

- Select the *Boundaries* Data Group.
- Click *Add* to create a boundary. We will use the default name ‘Boundary 1’.
- Set *Define boundary by* to *Orientation* which is default. This means that the boundary is considered along a full side of the computational grid. Specify the boundary orientation in the box *Boundary orientation* indicating on which side the boundary condition is applied. Select boundary orientation: *West*.
- Set the *Conditions along boundary* to *Space-varying* to indicate that the wave conditions vary along the up-wave boundary (see Figure 6.8).

➢ Click *Edit conditions* to specify the incident wave parameters at the selected boundary (note that the mean wave direction has to be in agreement with the convention specified in the sub-data group *Physical parameters – Constants* (i.e. Cartesian or Nautical convention)). Enter the wave parameters for the first section (see Figure 6.9):

Distance from corner point: 1500 m
Significant wave height: 0.0 m
Peak period T_p: 5.0 s
Direction (nautical): 255 degrees
Directional spreading: 4 [-]
Select *Counter clockwise*

➢ Click *Add* for a new section. Enter the wave parameters for the second section:

Distance from corner point: 9000 m
Significant wave height: 1.0 m
Peak period T_p: 5.0 s
Direction (nautical): 255 degrees
Directional spreading: 4 [-]
Select *Counter clockwise*

➢ Click *OK* to close the **Space-varying boundary conditions** window.

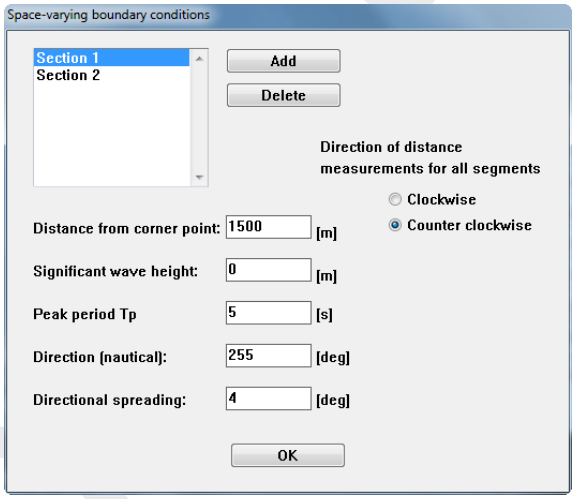

Figure 6.9: *Space-varying boundary conditions*

For the *Specification of spectra*, select *Parametric* to give the boundary conditions in a form of parametric input. To specify the spectral parameters that will be used:

➢ Click on *Edit spectral space*

The window **Spectral Space** will appear. A *JONSWAP* type spectrum will be used with the peak enhancement factor *Peak enh. fact.* set to the default value of "3.3". In the present tutorial the peak period *Peak* and the directional spreading expressed in *Degrees* are considered as input integral waves parameters. Select these options in the present window; see Figure 6.10.

➢ Click *OK* to confirm.

6.2.10 Obstacles

Within the Data Group *Obstacles* you can specify the characteristics of (a line of) sub-grid obstacles. The location of the obstacle is defined by a sequence of corner points of a line. The obstacles interrupt the propagation of the waves from one grid point to the next wherever this

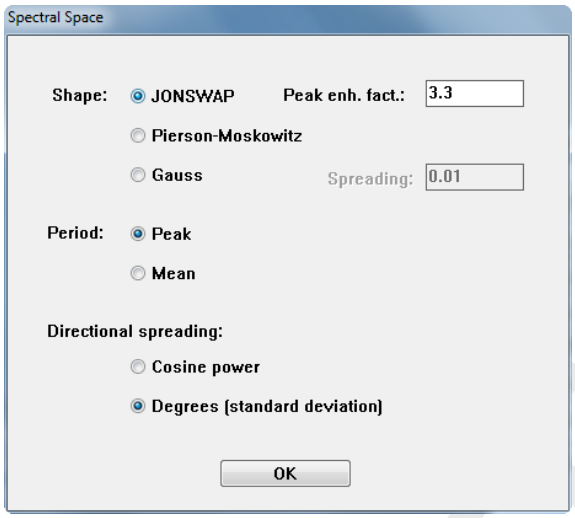

Figure 6.10: *Spectral space input parameters*

obstacle line is located between two neighbouring grid points of the computational grid (the resolution of transmission or blockage is therefore equal to the computational grid spacing).

- Click the Data Group *Obstacles*.
- Select *Add* to specify that an obstacle is present (this is the first obstacle).

Add may be used more than once to define more obstacles.

- Select *Dam* as obstacle type to specify that the transmission coefficient depends on the incident wave conditions at the obstacle and on the obstacle height (which may be submerged). The default values are used for reflection ("no") and for the *Height* of the dam (with respect to the reference level) and the coefficients *Alpha* and *Beta*.
- Select *Add* from the *Obstacle segment* item.
- Enter the co-ordinates of the first corner point of the obstacle in the *X-start* and *Y-start* boxes. Enter x = "814800" [m] and y = "818000" [m].
- Enter the co-ordinates of the second corner point of the obstacle in the *X-end* and *Y-end* boxes (x = "814800" [m] and y = "820000" [m]) and click on any edit box to confirm. The first segment has now been specified, see Figure 6.11.

The button *Add* may be used more than once to include for more segments. You can click *Delete* to remove a selected obstacle or a segment from the list.

6.2.11 Physical parameters

In the Data Group *Physical parameters* you can specify a number of physical parameters. The following sub-data groups are available.

Constants	In this sub-data group you can assign values to some general parameters.
Wind	Here you can specify the wind conditions.
Processes	In this sub-data group you can select the physical processes in SWAN (i.e. type of formulation, dissipation processes, non-linear wave-wave interactions, diffraction).
Various	Here you can switch on or off wave propagation in spectral space and several physical processes in SWAN.

DRAFT

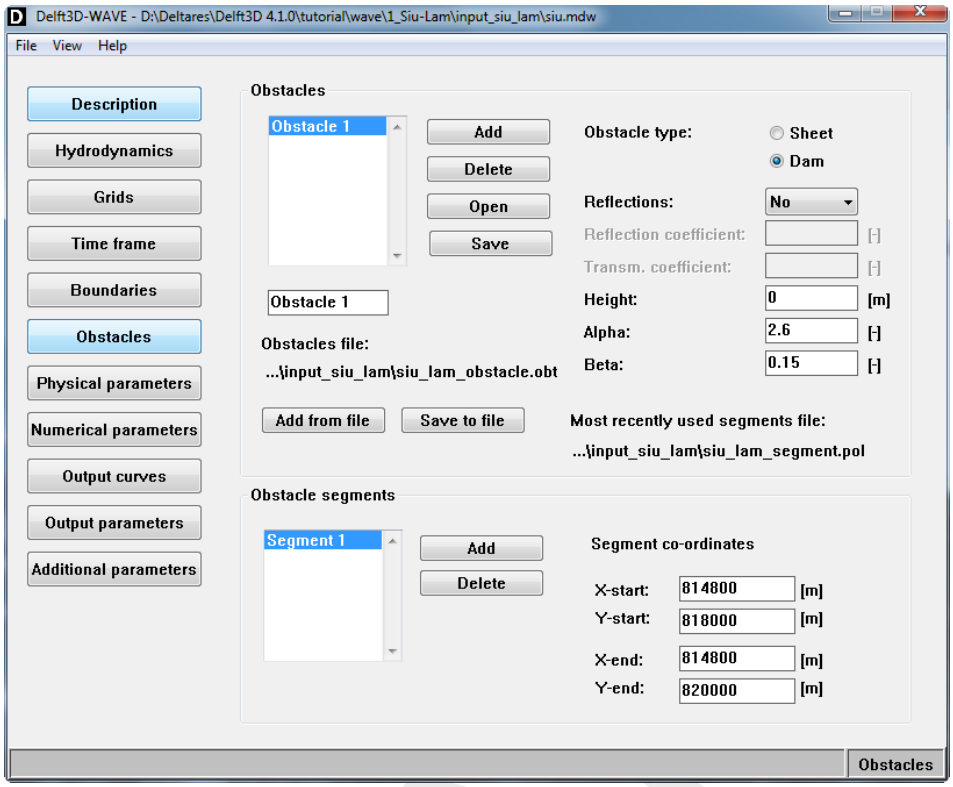

Figure 6.11: *Data Group* Obstacles

6.2.11.1 Constants

➢ Click the Data Group *Physical parameters* to show the sub-data groups.
➢ Select *Constants* in order to assign values to various general input parameters, see Figure 6.12.

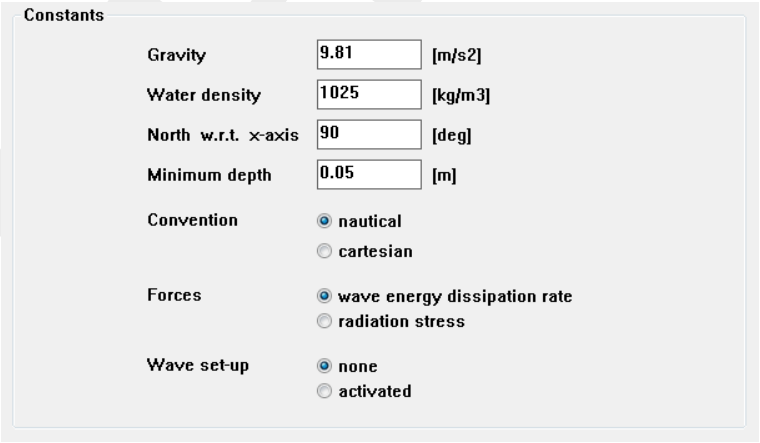

Figure 6.12: *Sub-data Group* Constants

The standard values for the gravitational acceleration *Gravity*, the *Water density*, the direction of *North* with respect to the x-axis and the threshold depth (in m) *Minimum depth* will be used. The *nautical* convention for wind and wave direction (button input and output) will be adopted in this tutorial. *Wave set-up* (within the SWAN model!) is de-activated. The *Forces* will be based on the *wave energy dissipation rate*.

6.2.11.2 Wind

➢ Select *Wind* to specify the wind conditions, see Figure 6.13.

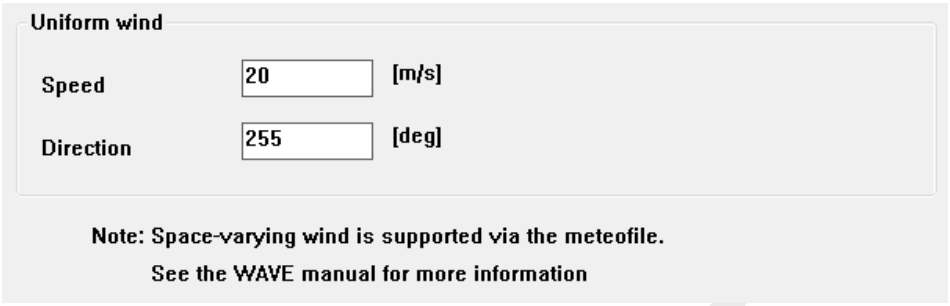

Figure 6.13: *Sub-data Group:* Wind

Here use will be made of a constant wind field (wind speed and direction). The wind direction applied is the same direction as the incident wave direction at the up-wave boundary.

➢ Enter the uniform wind parameters:
Speed: “20” m/s.
Direction: “255” degrees (conform the convention activated: Nautical)
➢ Select the Sub-data Group *Processes*. Next the window in Figure 6.14 is displayed.

6.2.11.3 Processes

In this sub-data group the physical processes to be activated in SWAN can be selected. Type of formulations: you can specify the mode in which SWAN can operate (first-, second-, third-generation mode).

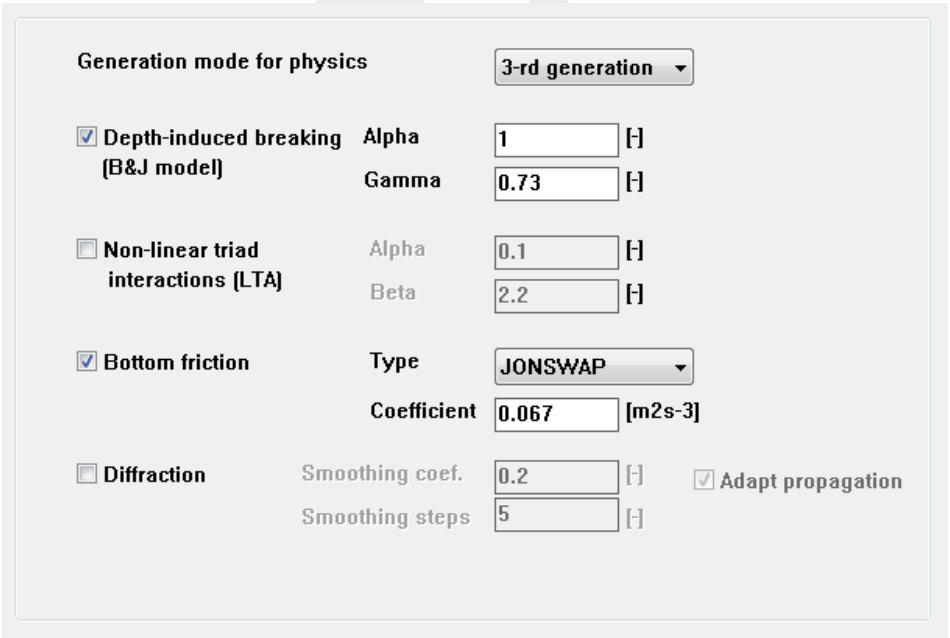

Figure 6.14: *Sub-data Group* Processes

➢ Select *3–rd generation* mode.

This means that SWAN will use third-generation formulations for the representation of the deep water physical processes.

Within the **Depth-induced breaking** sub-window you can activate depth-induced wave breaking using B&J model. Here the default values are used (*Alfa* = 1.0 and *Gamma* = 0.73).

Within **Non-linear triad interactions (LTA)** sub-window you can activate the triad wave-wave interactions based on the *LTA* (i.e. Lumped Triad Approximation, see section 4.5.7 and section 7.4.3) with default values for *Alfa* = 0.10 and *Beta* = 2.2. De-activate the *Non-linear triad interactions (LTA)*.

➢ To activate dissipation by bottom friction, check *Bottom friction* (see section 4.5.7 and section 7.4.2). Select the *JONSWAP* bottom friction formulation with its default value 0.067 m^2/s^3.

Within the **Diffraction** sub-window you can activate diffraction. De-activate *Diffraction*.

6.2.11.4 Various

Within the Sub-data Group *Various* (see Figure 6.15) you can de-activate or activate several physical processes in order to perform, e.g. a sensitivity study. Keep all processes activated.

◇ When wind is present, the quadruplets are activated when using the third-generation mode for physics.

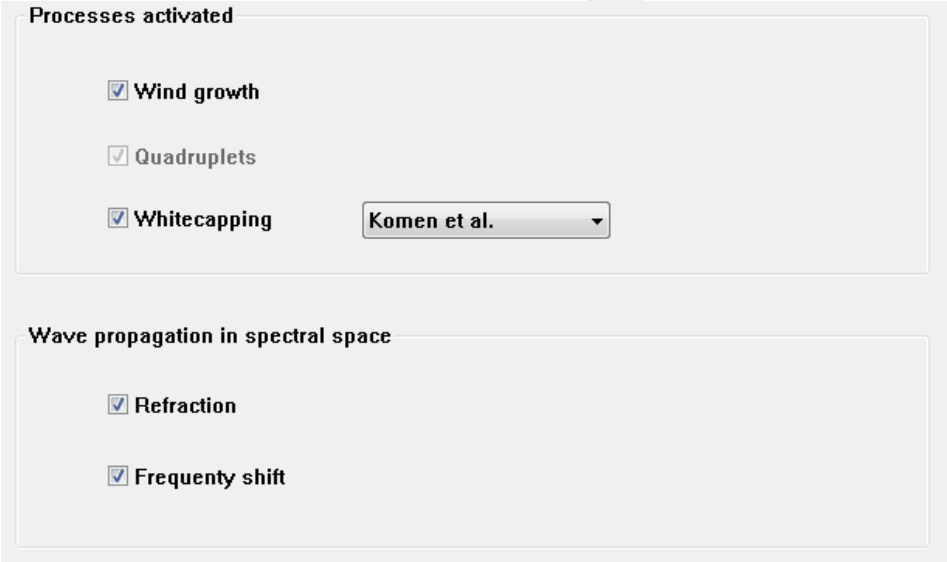

Figure 6.15: *Sub-data Group* Various

Remarks:

◇ For initial SWAN runs, it is strongly advised to use the default values of the model coefficients.

◇ Switching off depth-induced breaking is usually not recommended, since this leads to unacceptably high wave heights near beaches (the computed wave heights 'explode' due to shoaling effects).

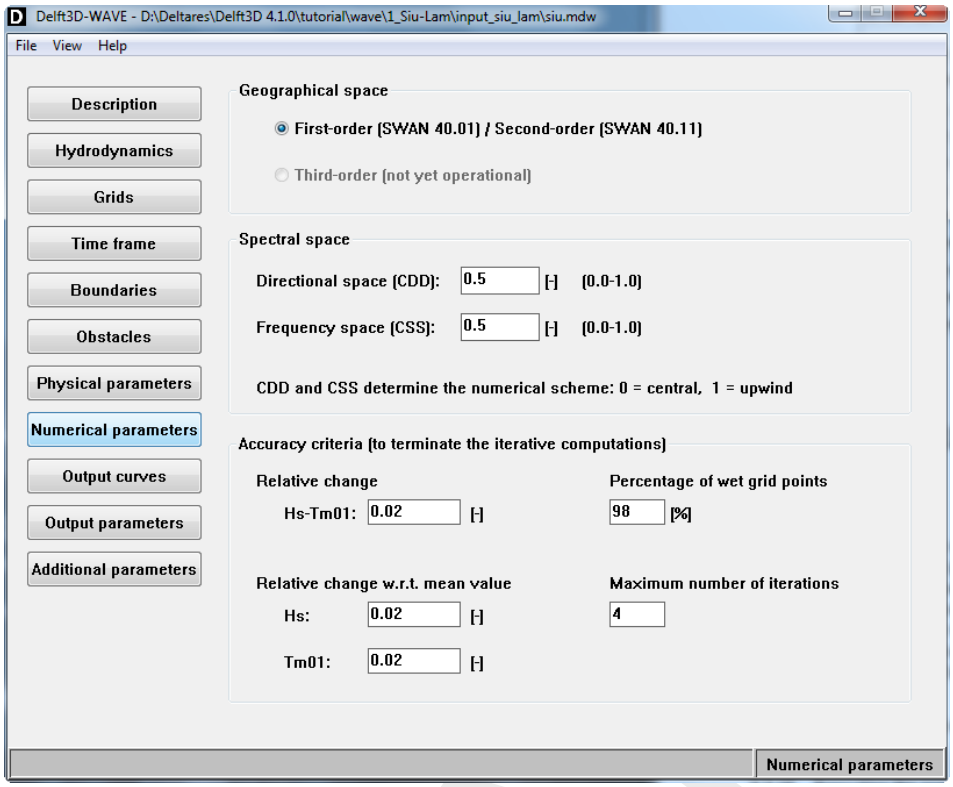

Figure 6.16: *Data Group* Numerical parameters

6.2.12 Numerical parameters

In the Data Group *Numerical parameters* you can modify parameters that affect the stability and accuracy of the numerical computation.

➢ Click the Data Group *Numerical parameters*: Next the window in Figure 6.16 is displayed.

In the **Spectral space** canvas you can control the amount of diffusion of the implicit scheme in the directional space through the parameter for the *Directional space (CDD)* and frequency space through the parameter for the *Frequency space (CSS)*. The default values will be used here.

In the canvas **Accuracy criteria (to terminate the iterative computations)**, you can influence the criteria for terminating the iterative procedure in the SWAN computations (for convergence criteria of SWAN see section 4.5.8). Here the default values are used for the *Relative change*, the *Relative change w.r.t. mean value* and the *Percentage of wet grid points*.

You can also specify the *Maximum number of iterations* at which the computation stops.

➢ In this tutorial change the default number of “15” iterations into “4” iterations.

6.2.13 Output curves

Within the Data Group *Output curves* you can specify an output curve at which wave output should be generated by Delft3D-WAVE. Actually the curve is piecewise linear. In this tutorial no output curves will be defined.

6.2.14 Output parameters

Within the Data Group *Output parameters* (see Figure 6.17) you can determine to which grid (i.e. wave or flow grid) output is written and to which extent the computations should be monitored. The latter option can be used to specify that Delft3D-WAVE should produce intermediate (model) results during a SWAN run (test output) if the program produces unexpected results.

Within this data group it is also possible to select output locations for which Delft3D-WAVE produces wave output that is directly obtained from SWAN, e.g. 2D wave spectra.

➢ Select *Output parameters* to enter the Data Group (Figure 6.17).

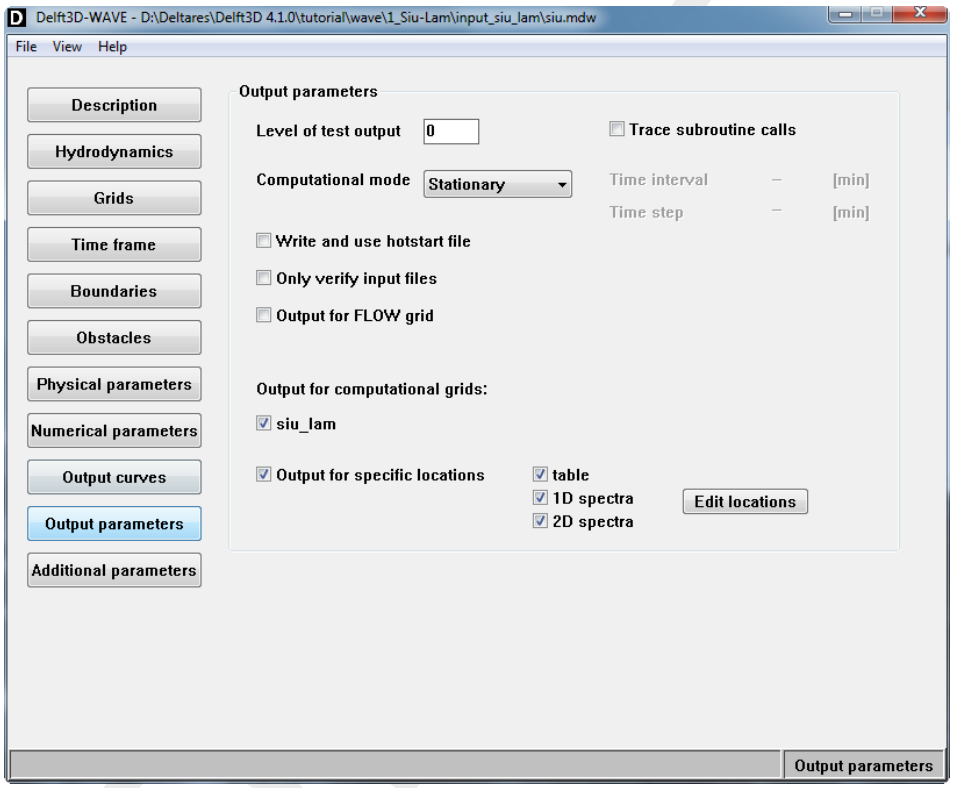

Figure 6.17: *Data Group* Output parameters

The default values for *Level of test output* and *Debug level* will be used. No hotstart file will be written and used.

Sometimes we want as much results as possible, Delft3D-WAVE offers to save the results of the calculation: on the communication file <com-∗.dat> and on a SWAN output file <wavm-∗.dat>. An overview of these output files is given in chapter 5.

➢ Check *Output for computational grids* button *siu_lam* to save the results on the <wavm-∗.dat> output file
➢ and check *Output for specific locations* to indicate that SWAN output should be generated at some locations.
➢ Click on *Add* to edit the x- and y-co-ordinates of the output locations. Enter the location with co-ordinates (826000, 823000), see Figure 6.18.
➢ Save file by pressing the *Save* button.
➢ Press the *Close* button.

For the selected location you can have three types of output: *table*, *1D spectra*, *2D spectra*.

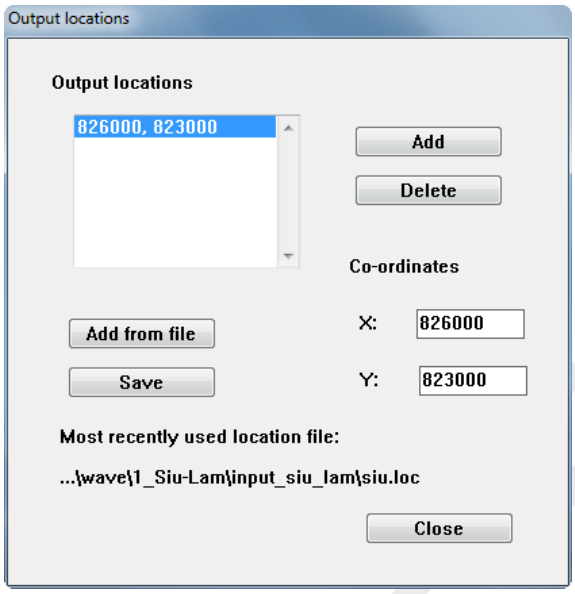

***Figure 6.18: Output locations** window*

➢ Select all these options for the Siu-Lam case.

Remarks:

- ◇ The Table output for specific locations is stored in <*case*.tab> for the overall computational grid, <*case*ni.tab> for the i–th nested grid.
- ◇ The 1D spectra output for specific locations is stored in <*case*.sp1> for the overall computational grid, <*case*ni.sp1> for the i–th nested grid.
- ◇ Similar for the 2D spectra output in <∗.sp2> files.

➢ After the input is completed, select *File* → *Save As* to save the input as <siu.mdw> file.
➢ Select *File* → *Exit* to close the WAVE-GUI.

Now the scenario is ready to be executed.

6.2.15 Additional parameters

Description of *Additional Parameters* is under construction

6.2.16 Executing the scenario

A wave scenario is stored in an <∗.mdw> file. To execute a wave scenario, the <∗.mdw> file must be selected.

➢ To start in foreground, select *Start* in the **Wave (standalone)** menu.
➢ Select the WAVE input file <siu.mdw> (see Figure 6.19).

DRAFT

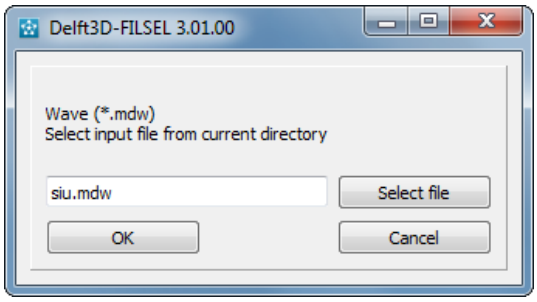

Figure 6.19: *Select scenario to run*

➢ Confirm by *OK* and the wave computation will be carried out.

In foreground the status of the simulation and possible messages are displayed in the active window. The simulation will start. After the simulation has finished check the results with the postprocessing program.

After the simulation is finished you are strongly advised to inspect at least some of the report files generated during the simulation to check if all went according to plan. This concerns especially the <swn-diag> file. To see this report:

➢ Select *Report* in the Delft3D-MENU window to inspect the report file of the wave model SWAN.

Especially the end of the <swn-diag.siu> file is of importance as it summarises errors, warnings and information of the computation.

6.2.17 Output files of Delft3D-WAVE

The result files of the calculation are of the NEFIS file format. The result files are:

<wavm-siu.dat> and <wavm-siu.def>

The results of the calculation can be visualised using either GPP or Delft3D-QUICKPLOT postprocessors. A description of the output parameters available on the output files is given in section 5.3.2.

6.2.18 Visualising results

The results presented in this section are generated using the GPP postprocessor.

In Figure 6.20 to Figure 6.24 some results are shown of the computed wave pattern near Siu-Lam. To reproduce these plots you should start the postprocessing program GPP.

➢ Select GPP either in the **Waves** window or in the **Utilities** window of Delft3D-MENU.
➢ In the main window of GPP select *Session* - *Open*.
➢ In the file selection menu select and open the session file <tutorial_swan_siu_lam.ssn>.
➢ In the main window of GPP select *Plots* and select from the list of possible plots the one you would like to inspect.

In an <∗.ssn> file the references are stored to data sets, in this case the result files of the siu-scenario, and the definition of earlier defined plots and their layout. By calling this scenario file you can inspect the same plots after repeating the simulation with (other input data) of the WAVE scenario “siu”. For details of using GPP you are referred to the User Manual of GPP.

DRAFT

To return to the main window of GPP while viewing a plot:

➢ Select *Plot* → *Close*.

You can select another plot as described above.

To close GPP and return to Delft3D-MENU:

➢ Select in the main window of GPP *Session* → *Exit*.

To close Delft3D-MENU:

➢ Select *Return*.
➢ Select *Exit*.

6.3 Nested wave model

This tutorial discusses the set-up of a wave model in which a nesting procedure takes place for a specific example called 'Friesian Inlet'. The modelled area covers an area in the north of The Netherlands called the 'Wadden Sea', which is an open sea protected by a series of barrier islands. Most of the input definition is already discussed in Tutorial 1 and therefore we will mention those steps briefly. Only the additional steps needed for the wave simulation are presented in this tutorial.

It is noted that the used wind, wave and other parameters do not represent realistic conditions for that area. Therefore the presented results have no practical use.

The input data is located on the directory

<. . . /tutorial/wave/2_Nested_wave_model/input_nested_wave>

and need to be copied first to the directory

<. . . /tutorial/wave/2_Nested_wave_model/>.

The files used in the case of *Friesian Inlet* are:

<netherlands.ldb>	land boundary file
<wadden_sea.grd>	Delft3D grid file
<wadden_sea.enc>	Delft3D enclosure file
<wadden_sea.dep>	Delft3D depth file
<inlet.grd>	Delft3D grid file
<inlet.end>	Delft3D enclosure file
<inlet.dep>	Delft3D depth file
<detailed.grd>	Delft3D grid file
<detailed.enc>	Delft3D enclosure file
<detailed.dep>	Delft3D depth file

6.3.1 WAVE Graphical User Interface

Start the WAVE-GUI on the directory <. . . /tutorial/wave/2_Nested_wave_model/> (see chapter 3 for details).

DRAFT

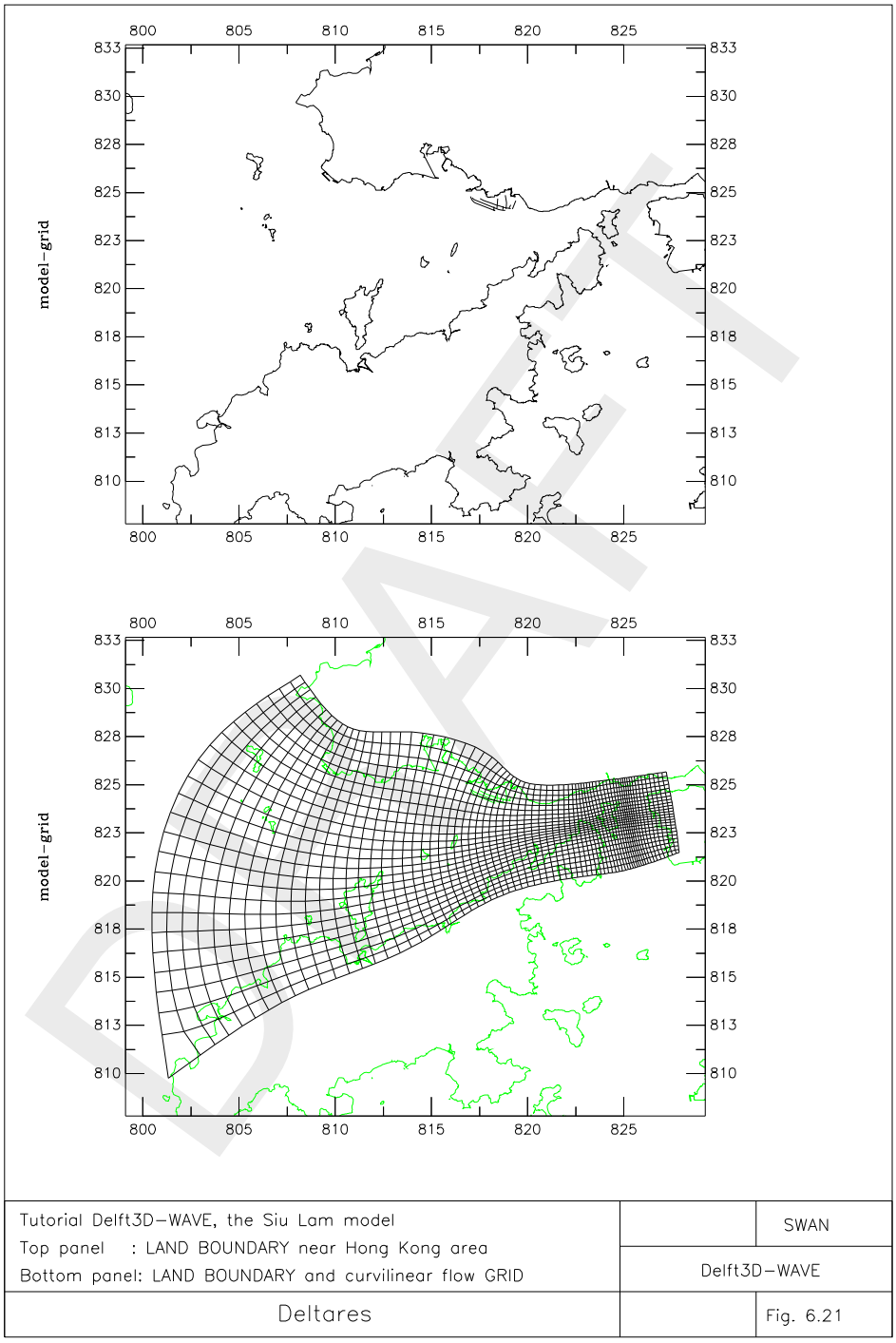
model-grid
800 805 810 815 820 825
833 830 828 825 823 820 818 815 813 810
model-grid
800 805 810 815 820 825
833 830 828 825 823 820 818 815 813 810
Tutorial Delft3D-WAVE, the Siu Lam model
Top panel : LAND BOUNDARY near Hong Kong area
Bottom panel: LAND BOUNDARY and curvilinear flow GRID
SWAN
Delft3D-WAVE
Deltares
Fig. 6.21

Figure 6.20: *Top panel: Siu Lam model area near Hong Kong area.*
Bottom panel: LAND BOUNDARY and curvilinear flow GRID

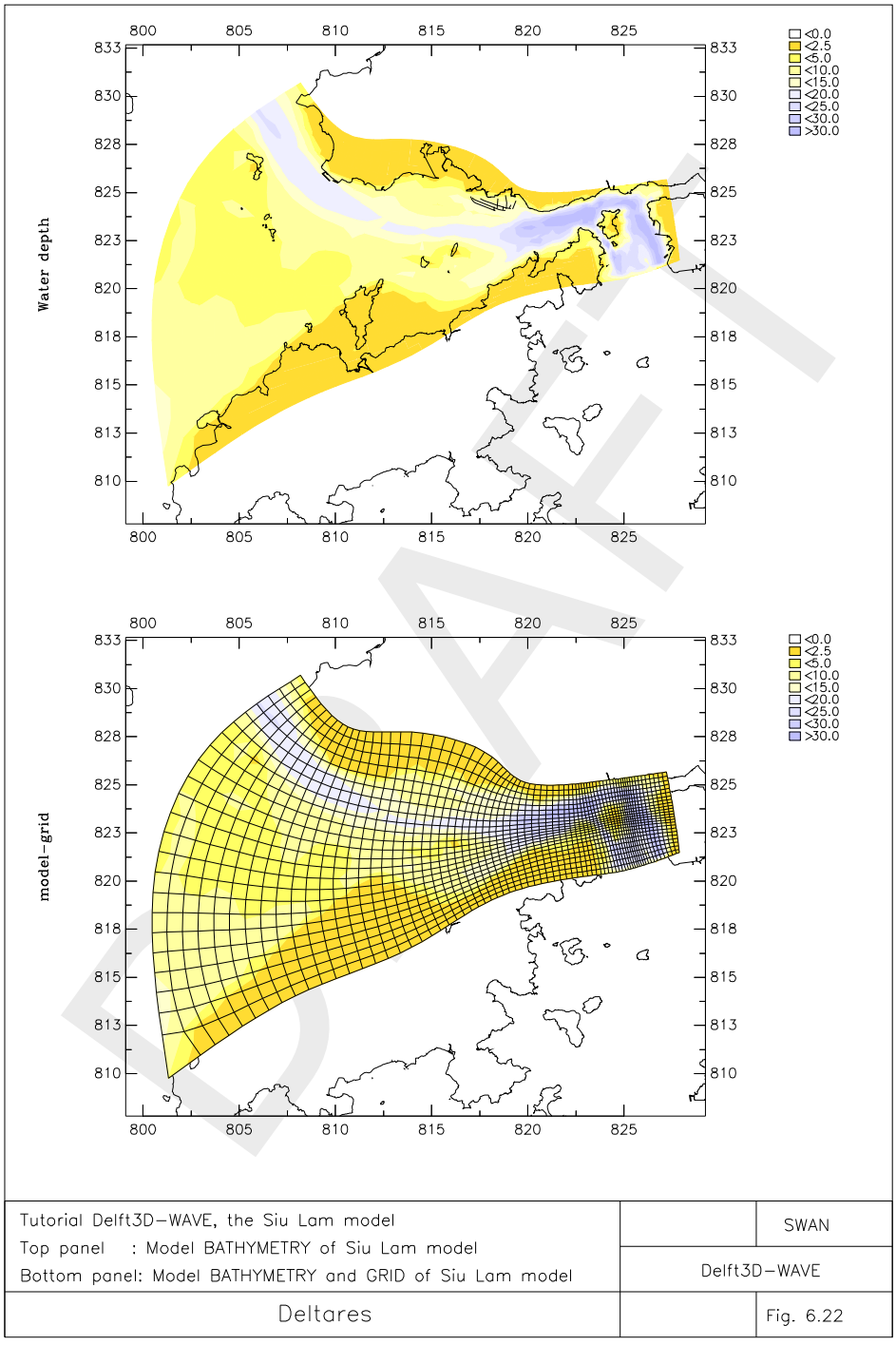

Figure 6.21: *Top panel: Model BATHYMETRY of Siu Lam model.*
Bottom panel: BATHYMETRY and GRID of Siu Lam model

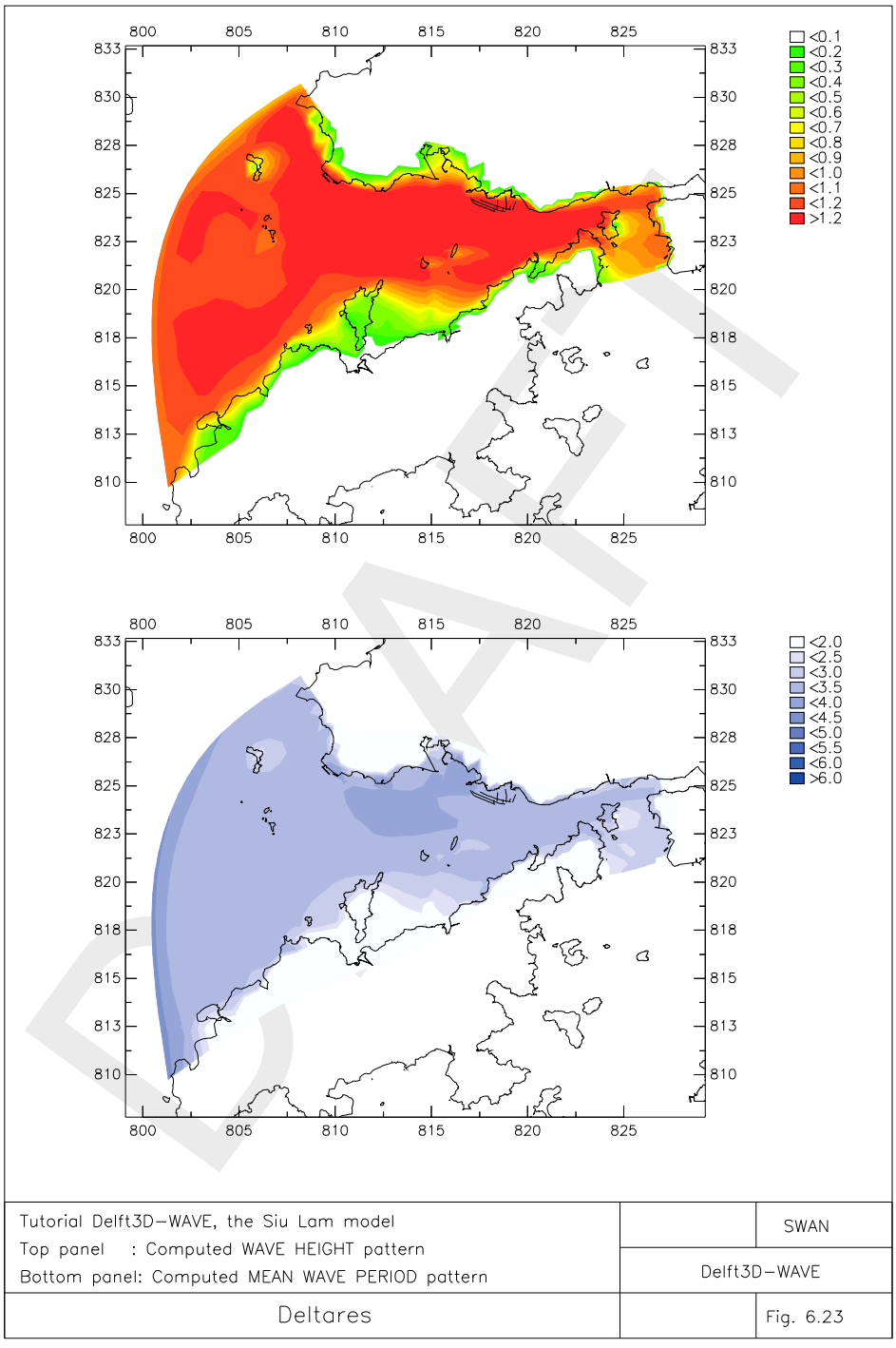

Figure 6.22: *Top panel: Computed WAVE HEIGHT pattern on 1 Oct 2005, 18:00. Bottom panel: Computed MEAN WAVE PERIOD pattern on 1 Oct 2005, 18:00*

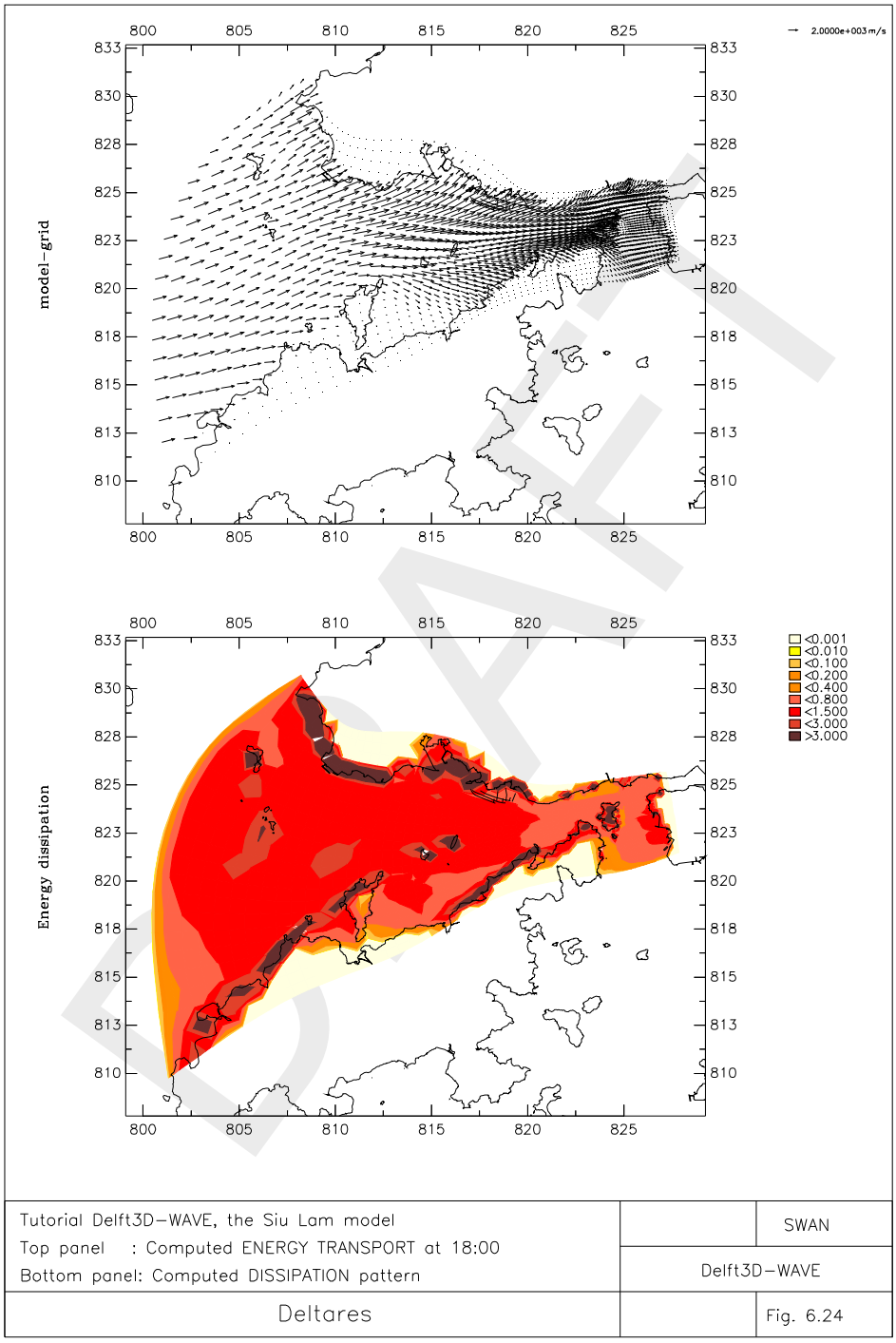

Figure 6.23: *Top panel: Computed ENERGY TRANSPORT on 1 Oct 2005, 18:00. Bottom panel: Computed DISSIPATION pattern on 1 Oct 2005, 18:00*

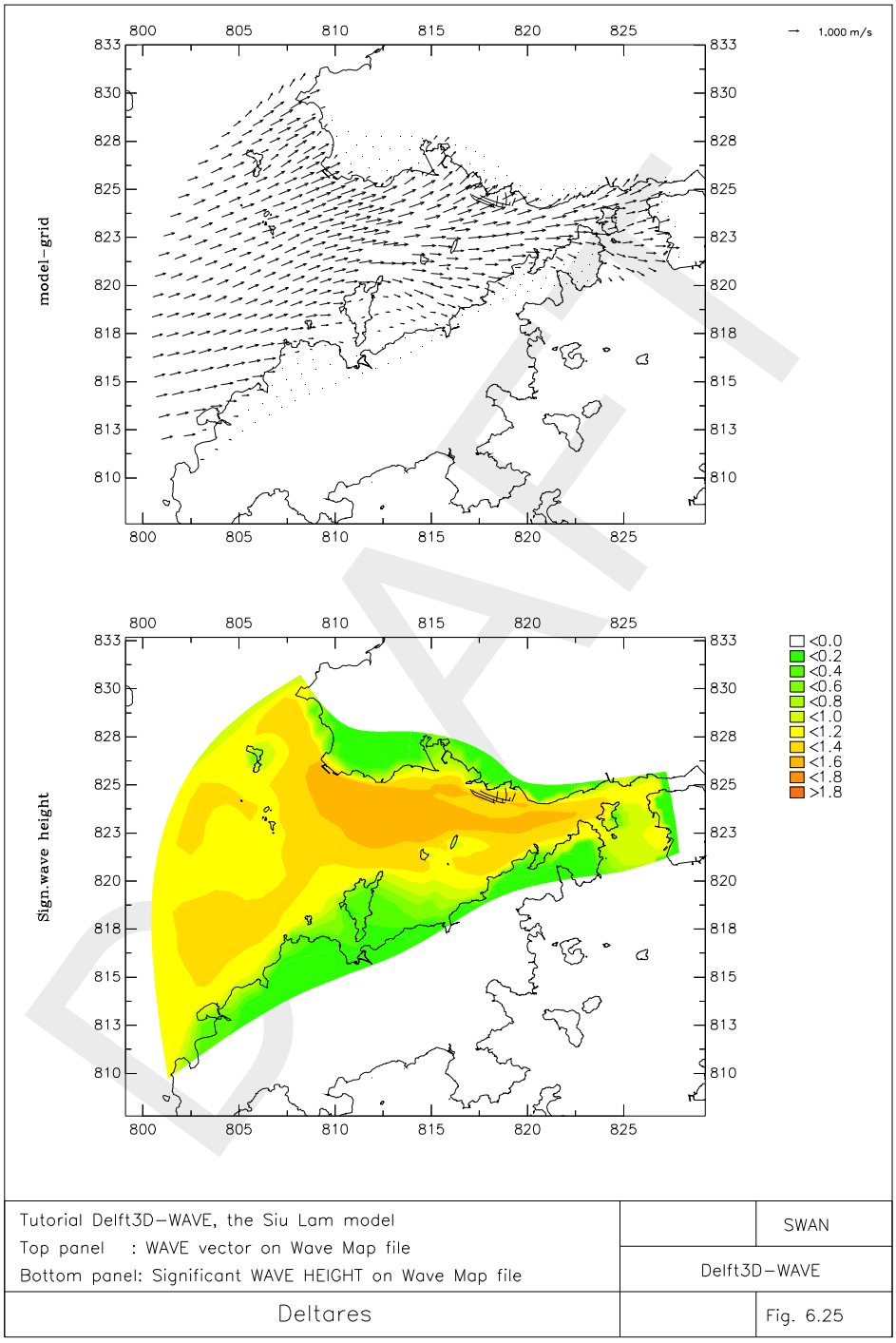

Figure 6.24: *Top panel: WAVE vector on 1 Oct 2005, 18:00.*
Bottom panel: Significant WAVE HEIGHT on 1 Oct 2005, 18:00

6.3.1.1 Description

Type the description:

- ➢ Project name: “Friesian Inlet”
- ➢ Project: “f01”
- ➢ Description: “Tutorial Delft3D-WAVE”
  “Friesian Inlet”
  “Standalone Wave model with nesting”

6.3.1.2 Hydrodynamics

No information from a flow model is used in this tutorial.

6.3.1.3 Grids

When you want to make nested runs, you first have to import all considered grids. In this canvas three computational grids must be imported.

- ➢ Start with importing the coarsest grid called <wadden_sea.grd>.

For this grid, a corresponding depth file must be imported, where the bathymetry data is based on the selected computational grid.

- ➢ Go to the tab **Bathymetry** and open the file <wadden_sea.dep>.
- ➢ Import the <inlet.grd> and the corresponding <inlet.dep> file.
- ➢ Finally, import the <detailed.grd> and its depth file <detailed.dep>.

In the tab **Nesting** you must define from which grid the selected grid must obtain its boundary conditions.

- ➢ First select the *detailed* grid in the list box for *Computational grids*.

You can select grids from the computational grid window by clicking on the presented grids. The line of that grid will become dark-blue (see Figure 6.25).

Note that for the last grid, a choice can be made between two grids (see Figure 6.25). It is possible to nest the <detailed.grd> in the <wadden_sea.grd> or in the <inlet.grd>.

- ➢ Select the <inlet.grd>.

Also check if the <inlet.grd> is nested in the <waddensea.grd>.

Remarks:

- ◇ The first grid cannot be nested in another one. For this grid, boundary conditions must be specified in the Data Group *Boundaries*.
- ◇ A grid cannot be nested in itself.
- ◇ If land points remain dry during the computations, then these points will be ignored for the SWAN computation.

6.3.1.4 Time frame

- ➢ Add one time point for the wave computation: “04 08 2005 00 00 00” [dd mm yyyy hh mm ss]

DRAFT

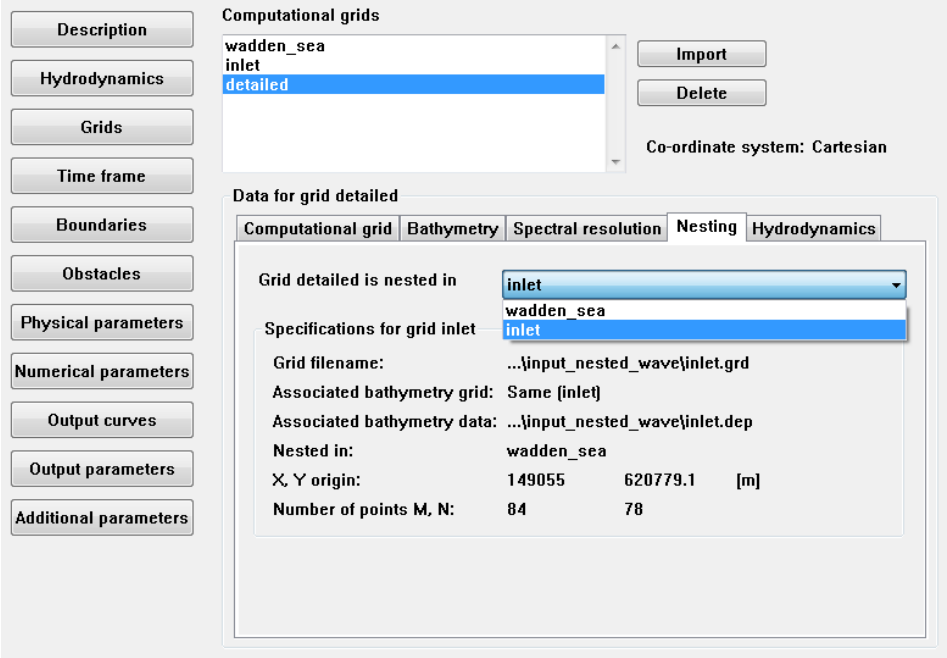

Figure 6.25: *Data Group Grids – Nesting window*

6.3.1.5 Boundaries

- Add one boundary called “Boundary West” with orientation ‘West’.
- Specify uniform conditions along the boundary with:

  Significant wave height: “2.0” m
  Peak period T_p: “6.3” s
  Direction (nautical): “270” degrees
  Directional spreading: “4” [-]

6.3.1.6 Obstacles

No obstacles are applied.

6.3.1.7 Physical parameters

- Press button *Wind*
- Define a uniform wind with a wind speed of “10” m/s coming from the southwest (225°).

6.3.1.8 Numerical parameters

Keep the default values.

6.3.1.9 Output curves

No output curves are defined.

6.3.1.10 Output parameters

- Select to write output to all three computational grids (see Figure 6.26).
- Save the input as <wad.mdw>.

DRAFT

Figure 6.26: *Data group* Output parameters*, output for computational grids*

6.3.1.11 Additional parameters

No additional parameters are defined.

6.3.2 Run and postprocessing

Execute this scenario and check the results with a postprocessing program.

You will see that SWAN will make three computations: one computation for each computational grid. The result files will be named as follows:

<wavm-wad-waddensea.dat> and <wavm-wad-waddensea.def>
<wavm-wad-inlet.dat> and <wavm-wad-inlet.def>
<wavm-wad-detailed.dat> and <wavm-wad-detailed.def>

The wave map files include the grid name in the name, so you can directly select the correct output file.

6.4 Online-WAVE coupling (including morphology)

6.4.1 Introduction

This tutorial discusses the set-up of a model with flow-wave interaction. The coupling of the FLOW and WAVE module is discussed based on an example called *Bornrif*. The modelling area concerns the Ameland inlet, which is a part of the Wadden Sea. The model was made (Wilkens, 1999) to reproduce the development of a spit at the head of Ameland: the *Bornrif* (see Figure 6.27). It appeared to be possible to reproduce the spit forming, though not with the schematisations of the real acted hydraulic conditions. Only wind and waves from the west or north-west were applied. The simulated time span amounted five years and took three days of computation time.

Figure 6.27: *Development of a spit at the Head of Ameland: the Bornrif*

The morphological development agreed partly with the observations. The spit is formed, but not with the correct shape. The severe erosion of the ebb delta and the sedimentation north and east of it are less extensive in reality. The migration of the channels is not reproduced very well.

The driving forces of the bathymetry development appeared to be the waves. This is correct to a certain extent. In this model, however, the sediment transport of waves seemed to be overestimated with respect to the currents. In areas sheltered from waves the morphological activity is too small. It was also concluded that the transport over the flats is probably too high with respect to the transport through the channels.

DRAFT

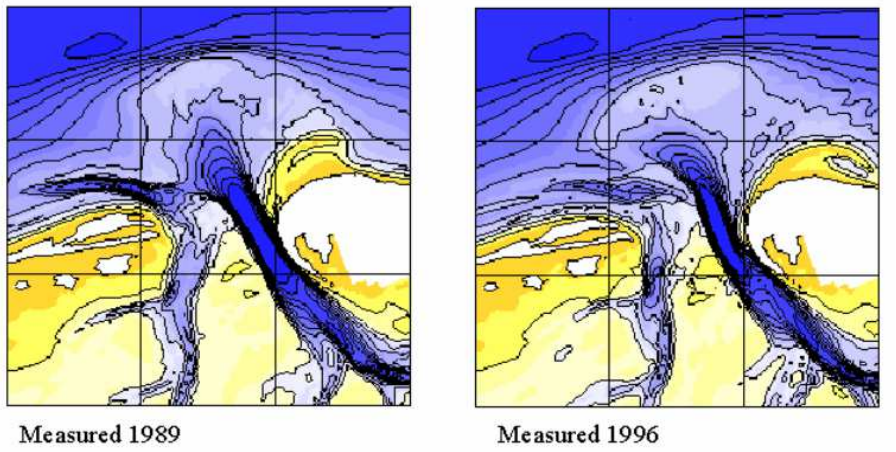

Figure 6.28: *Measured 1989 and 1996 bathymetry*

This tutorial covers a part of the above simulation focussing only on one wave condition. Also the morphodynamic aspects are taken into account in this tutorial.

The input data is located on the directory

<.../tutorial/wave/3_bornrif/input_bornrif>

and need to be copied first to the directory

<.../tutorial/wave/3_bornrif/>.

The files used in the *Bornrif* case are:

Delft3D-FLOW:

<netherlands.ldb>	Landboundary file
<rif.grd>	Delft3D grid file
<rif.enc>	Delft3D enclosure file
<rif.dep>	Delft3D depth file
<rif_neu.bnd>	Delft3D open boundaries file
<rif.bch>	Delft3D boundary condition file (harmonic)
<rif.bcc>	Delft3D boundary condition file (concentration)
<rif.wnd>	Delft3D wind file
<rif_200.sed>	Delft3D file containing sediment data
<rif_200.mor>	Delft3D file containing morphological data
<rif.obs>	Delft3D Observation points
<rif.crs>	Delft3D Cross-sections

Delft3D-WAVE:

<wave_overall.grd>	Delft3D grid file
<wave_overall.enc>	Delft3D enclosure file
<wave_overall.dep>	Delft3D depth file
<wave_detail.grd>	Delft3D grid file
<wave_detail.enc>	Delft3D enclosure file
<wave_detail.dep>	Delft3D depth file

To make a coupling between the FLOW and WAVE module, both flow- and wave-models

should be set up, before the coupling can be accomplished. First a description of the set-up of the FLOW model is discussed.

6.4.2 Delft3D-FLOW model

Start the FLOW-GUI on the directory <.../tutorial/wave/3_bornrif/> (see chapter 3 for details).

6.4.2.1 Description

Type a description of the project name, project identifier and a descriptive text:

- Name: "Friesian Inlet"
- Identifier: "f01"
- Text: "Tutorial Delft3D-WAVE"
  "Ameland Tidal Inlet"
  "Coupling of the FLOW and WAVE module"

6.4.2.2 Domain

- Import the flow grid <rif.grd>,
- the corresponding enclosure file <rif.enc> and
- related bathymetry file <rif.dep>.

No dry points and thin dams are specified here.

6.4.2.3 Time frame

- Choose as a reference date: "01 01 1996"
- Simulation start time: "01 01 1996 04 12 00"
- Simulation stop time: "01 02 1996 00 00 00"
- Time step: "1" minute

6.4.2.4 Processes

- In the FLOW datagroup *Processes*, the process *Wave* must be activated.

A window will appear with instructions related to the flow-wave coupling (see Figure 6.29). Read these instructions.

- Click *Go to Output* (see Figure 6.31).

It is also possible to do this in the end.

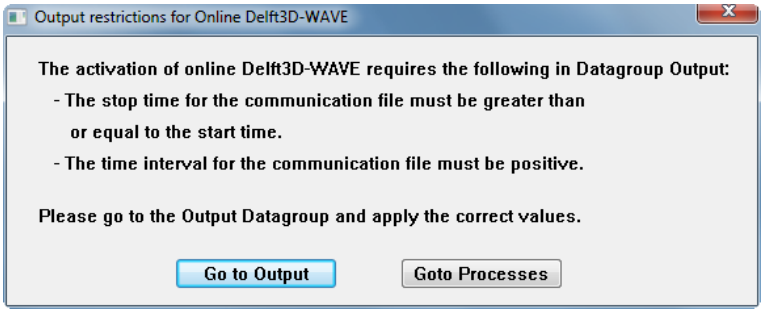
Output restrictions for Online Delft3D-WAVE
The activation of online Delft3D-WAVE requires the following in Datagroup Output:
- The stop time for the communication file must be greater than
or equal to the start time.
- The time interval for the communication file must be positive.
Please go to the Output Datagroup and apply the correct values.
Go to Output
Goto Processes

Figure 6.29: *Output restrictions for Online-WAVE*

The FLOW module now expects to read wave data from the communication file at certain

time points during the computation. The process *Online Delft3D-WAVE* should be selected to couple a Delft3D-FLOW computation directly with a Delft3D-WAVE computation. This feature is called the *Online-WAVE* option.

➢ Check *Online Delft3D-WAVE*

Two other processes (Sediments and wind) must be activated as well, which are flow related (see Figure 6.30):

➢ Check the buttonSediments
➢ Enter the name of non-cohesive sediment: “Sediment sand”
➢ Click on button *Add*
➢ To close the window click on button *Close*

➢ Check the checkbox *Wind*.

Now the window looks like Figure 6.30.

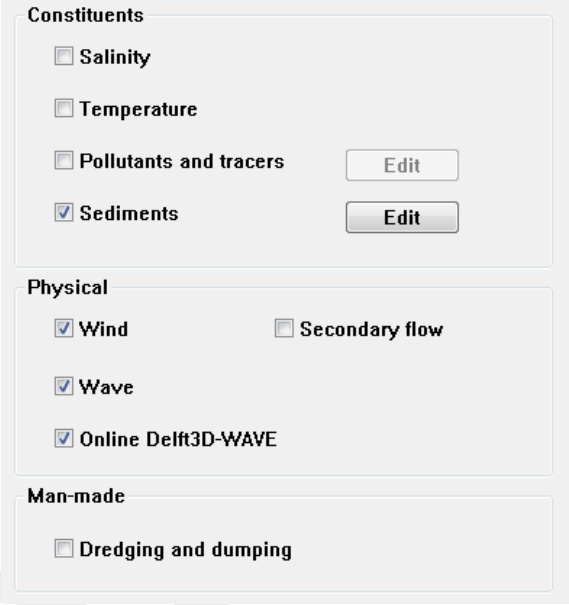

Figure 6.30: *Overview of active processes*

6.4.2.5 Initial conditions

➢ Enter “-0.45” m as the uniform value for water level.
➢ Enter “0” kg/m^3 for sediment sand.

6.4.2.6 Boundaries

➢ Press the button *Open*
➢ Import the following files:

<rif_neu.bnd>	boundary definitions,
<rif.bch>	harmonic boundary conditions and
<rif.bcc>	constituent boundary conditions.

➢ Press the *Close* button to close the window **Open/Save Boundaries**.

6.4.2.7 Physical parameters

Use all default values for the constants, but adapt wind drag coefficients

➢ Select tab *Constants*

Enter the following values for the *Wind drag coefficients*

➢ Breakpoint A: Enter coefficient “0.0025”, wind speed “0” m/s
➢ Breakpoint B: Enter coefficient “0.0025”, wind speed “100” m/s
➢ Breakpoint B: Enter coefficient “0.0025”, wind speed “100” m/s

➢ Select tab *Roughness*
➢ Set the uniform Manning roughness coefficient for both velocity components (U and V) to “0.026”.

Use the default values for viscosity.

➢ Select tab *Sediment*
➢ Press button *Open* and select the file <rif_200.sed>.

➢ Select tab *Morphology*
➢ Press button *Open* and select the file <rif_200.mor>.

➢ Select tab *Wind*
➢ Press button *Open* and select the file <rif.wnd>.

6.4.2.8 Numerical parameters

➢ Adapt the *Threshold depth* = “0.35” m

Use default for the other options

6.4.2.9 Operations

No operations (like dredge and dump) are specified in this case.

6.4.2.10 Monitoring

Import the files for observation points and cross sections. No drogues are specified in this case.

➢ Select button *Observations*
➢ Press button *Open* and select the file <rif.obs>.

➢ Select button *Cross-sections*
➢ Press button *Open* and select the file <rif.crs>.

6.4.2.11 Additional parameters

➢ Add the following keyword (see Delft3D-FLOW (2013)): “Cstbnd=#Yes#”

DRAFT

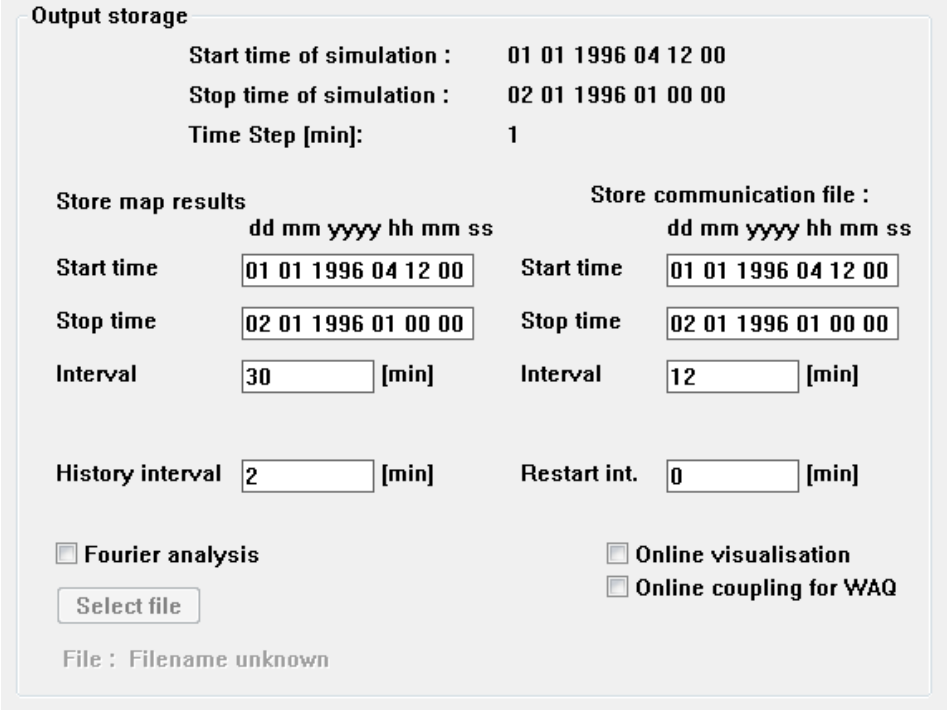
Output storage
Start time of simulation : 01 01 1996 04 12 00
Stop time of simulation : 02 01 1996 01 00 00
Time Step [min]: 1
Store map results
dd mm yyyy hh mm ss
Start time 01 01 1996 04 12 00
Stop time 02 01 1996 01 00 00
Interval 30 [min]
History interval 2 [min]
Store communication file :
dd mm yyyy hh mm ss
Start time 01 01 1996 04 12 00
Stop time 02 01 1996 01 00 00
Interval 12 [min]
Restart int. 0 [min]
Fourier analysis
Select file
File : Filename unknown
Online visualisation
Online coupling for WAQ

Figure 6.31: *Overview of output parameters*

6.4.2.12 Output

Store map results and prescribe the history interval as presented in Figure 6.31.

Essential for the flow-wave coupling is the storing interval of the communication file.

➢ At each interval (“12” minutes), the FLOW module will be updated with wave data starting at “01 01 1996 04 12 00” till the Stop time “02 01 1996 01 00 00”.

➢ Save this file as <rif.mdf>.

Exit the FLOW-GUI:

➢ Click *File* → *Exit.*

6.4.3 Delft3D-WAVE model

Start the WAVE-GUI on the directory <.../tutorial/wave/3_bornrif/> (see chapter 3 for details).

6.4.3.1 Description

Type the description:

➢ Project name: “Bornrif”
➢ Project: “003”
➢ Description: “Tutorial Delft3D-WAVE”
“FLOW-3DMOR and Online WAVE simulation”

DRAFT

6.4.3.2 Hydrodynamics

Select the hydrodynamic results from Delft3D-FLOW

- Check checkbox *Run WAVE together with FLOW*
- Select FLOW file <rif.mdf>.

6.4.3.3 Grids

In this exercise, the detailed grid is nested in an overall grid as shown in the section 6.3.

Computational grid

- Import the overall grid file: <wave_overall.grd>.
- Select tab *Bathymetry*.
- Import the related bathymetry file <wave_overall.dep>.
- Import the detailed grid file: <wave_detail.grd>.
- Import the related bathymetry file <wave_detail.dep>.

This grid must be nested in the overall grid. Use for both grids the spectral resolution as default values.

Hydrodynamics

- Select tab *Hydrodynamics*.

Select for both grids the items as follows:

- **Water level** option “Use but don’t extend”.
- **Current** option “Use but don’t extend”.
- **Bathymetry** option “Use but don’t extend”.
- **Wind** option “Don’t use”.

6.4.3.4 Time frame

The time frame is automatically read from the file <rif.mdf>. The storing interval to the communication file (i.e. 12 minutes) determines when Delft3D-WAVE is executed. Leave the water level correction on its default value.

6.4.3.5 Boundaries

Create the three boundaries North, East and West. First the North boundary.

- Press the *Add*
- Set *Boundary name* to “Boundary North”
- Set *Define boundary by* to “Orientation”
- Set *Boundary orientation* to “North”
- Set *Conditions along boundary* to “Uniform”
- Set *Specification of spectra* to “Parametric”
- Press button *Edit conditions*

- Significant wave height: “2.0” m
- Peak period T_p: “7.0” s
- Direction (nautical): “330” degrees
- Directional spreading: “4” [-]

The boundaries East and West use the same values, accept for the word ’North’ (0°).

DRAFT

6.4.3.6 Obstacles

No obstacles are defined.

6.4.3.7 Physical parameters

- Press button *Wind*
- Specify a uniform wind speed of “6” m/s and
- specify a wind direction of “330” degrees.

Use default settings for all other processes.

6.4.3.8 Numerical parameters

Keep the default values.

6.4.3.9 Output curves

No output curves are defined.

6.4.3.10 Output parameters

- Select *Write and use the hotstart file*.
- Select *Output for FLOW grid*, for writing the wave data to the communication file.
- Select *Output for computational grids* to write output for the computational grid <wave_overall.grd> and
- Select *Output for computational grids* to write output for the computational grid<wave_detail.grd> (see Figure 6.32).

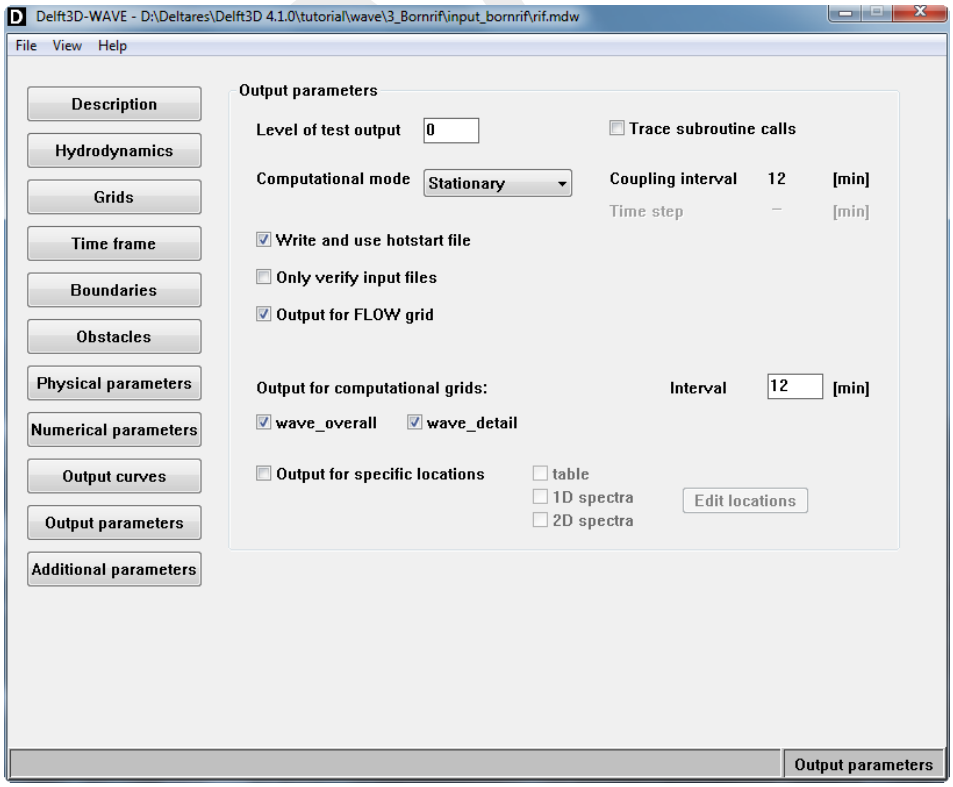

Figure 6.32: *Overview of output parameters in Delft3D-WAVE*

- Save the wave input to file <rif.mdw>.

Exit the WAVE-GUI:

➢ Click *File* → *Exit*.

6.4.4 Run and postprocessing

Executing this flow-wave model (including sediment and morphology) can be done by in foreground or background.

6.4.4.1 Foreground

To start in foreground

➢ Select *Start* in the **Hydrodynamics (including morphology)** menu.
➢ Select the FLOW input file <rif.mdf>.
➢ Confirm the selection by pressing *OK*.
➢ Select the WAVE input file <rif.mdw>.
➢ Confirm by *OK* and the flow-wave computation will be carried out.

The simulation will start. After the simulation has finished check the results with the postprocessing program.

6.4.4.2 Background

Go back to the main Delft3D menu

➢ Click *Batch* (see Figure 6.33).
➢ Click *Prepare*.
➢ Click *Start*.

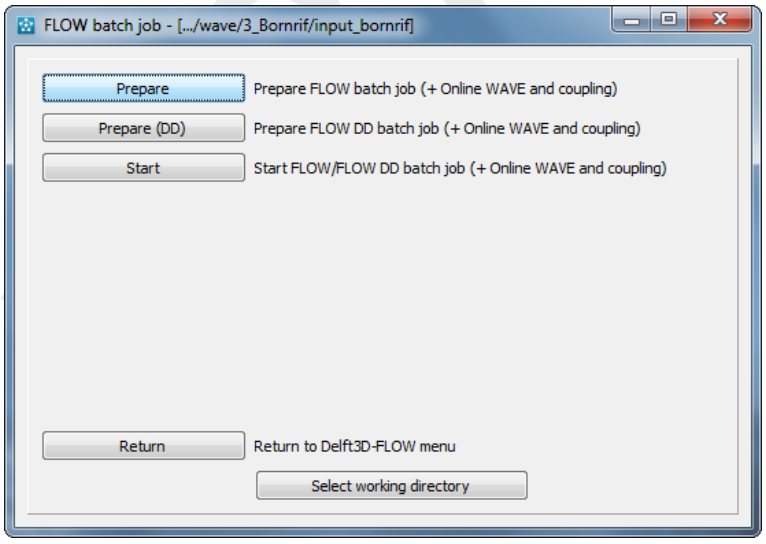
FLOW batch job - [.../wave/3_Bornrif/input_bornrif]
Prepare
Prepare FLOW batch job (+ Online WAVE and coupling)
Prepare (DD)
Prepare FLOW DD batch job (+ Online WAVE and coupling)
Start
Start FLOW/FLOW DD batch job (+ Online WAVE and coupling)
Return
Return to Delft3D-FLOW menu
Select working directory

Figure 6.33: *Execute the Flow-Wave model*

6.4.4.3 Output files

The FLOW module will create the following files:

<trim-rif.dat> and <trim-rif.def>
<trih-rif.dat> and <trih-rif.def>
<com-rif.dat> and <com-rif.def>.

The WAVE module will create the files:

<wavm-rif-wave_overall.dat> and <wavm-rif-wave_overall.def>
<wavm-rif-wave_detail.dat> and <wavm-rif-wave_detail.def>

The wave map files include the grid name in the file name, so you can directly select the correct output file.

6.5 FLOW-DD and Online WAVE

6.5.1 Introduction

In this tutorial the set-up of a Domain Decomposition model in combination with a Delft3D-FLOW-WAVE simulation is discussed, based on an example called Bornrif. With Domain Decomposition it is possible to divide the large domain into several smaller sub-domains. More information on Domain Decomposition can be found in Appendix B.13 of the Delft3D-FLOW user manual. The intention of this tutorial is only to illustrate the set-up of a domain decomposition model in combination with Delft3D-FLOW and Delft3D-WAVE. There is no physical functionality of domain decomposition in the Bornrif example. The domain decomposition is applied in the tidal channel between Terschelling and Ameland in order to obtain a higher resolution (factor 5) at that specific location. See Figure 6.34.

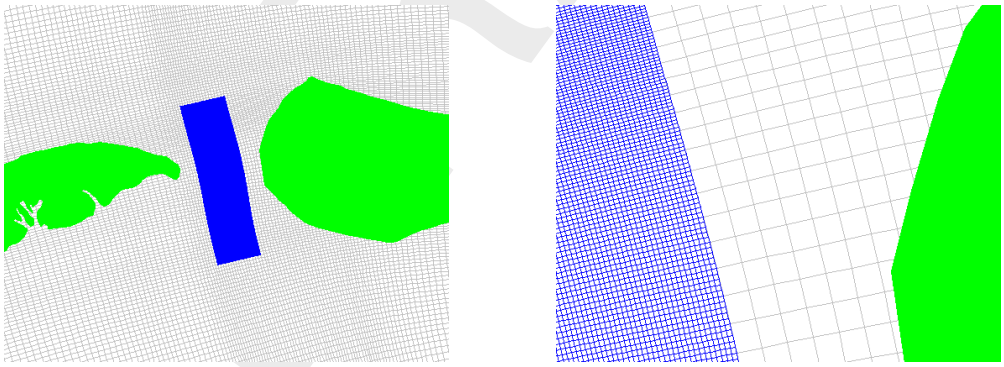

Figure 6.34: *Location of grids of both domains between Terschelling and Ameland (left panel) and detail of both domains close to Ameland (right panel)*

The input data is located on the directory

<.../tutorial/wave/4_bornrif_dd/input_bornrif>

and need to be copied first to the directory

<.../tutorial/wave/4_bornrif_dd/>.

The following files are used in the *Bornrif DD* case:

Delft3D-FLOW:

<netherlands.ldb>	Landboundary file
<rif_inside.grd>	Delft3D grid file
<rif_inside.enc>	Delft3D enclosure file
<rif_inside.dep>	Delft3D depth file
<rif_outside.grd>	Delft3D grid file
<rif_outside.enc>	Delft3D enclosure file
<rif_outside.dep>	Delft3D depth file
<rif_neu.bnd>	Delft3D open boundaries file
<rif.bch>	Delft3D boundary condition file (harmonic)
<rif.bcc>	Delft3D boundary condition file (concentration)
<rif.wnd>	Delft3D wind file
<rif_200.sed>	Delft3D file containing sediment data
<rif_200.mor>	Delft3D file containing morphological data

Delft3D-WAVE:

<wave_overall.grd>	Delft3D grid file
<wave_overall.enc>	Delft3D enclosure file
<wave_overall.dep>	Delft3D depth file
<wave_detail.grd>	Delft3D grid file
<wave_detail.enc>	Delft3D enclosure file
<wave_detail.dep>	Delft3D depth file

DD-boundaries:

<inside_outside.ddb>	DD-boundary file for coupling of both grids

Other files are generated during the tutorial. For the simulation it is advised to store all the files in the same directory.

The complete model set-up of a FLOW-DD and Online WAVE simulation is discussed below. A description about the construction and coupling of sub-domain grids can be found in the RGFGRID and FLOW user manuals (RGFGRID, 2013; Delft3D-FLOW, 2013). The model set-up starts with a description of the Delft3D-FLOW models.

6.5.2 Delft3D-FLOW models

For each sub-domain an mdf-file must be specified. First the set-up of the outside domain is discussed, followed by the set-up of the inside domain

6.5.2.1 Model set-up outside FLOW domain

Start the FLOW-GUI on the directory <.../tutorial/wave/4_bornrif_dd/>.

6.5.2.2 Description

➢ Type the description:

"Tutorial Delft3D-WAVE"
"Ameland Tidal Inlet"
"Combining FLOW-DD and WAVE"
"Outside model"

DRAFT

6.5.2.3 Domain

- Select tab *Grid*
- Import grid file <rif_outside.grd> and grid enclosure file <rif_outside.enc>
- Set the latitude to “52” degrees.
- Select tab *Bathymetry*
- Import the corresponding bathymetry file <rif_outside.dep>.

Dry points and thin dams are not specified in this case

6.5.2.4 Time frame

Set the following timings:

- Reference date: “01 01 1996”
- Simulation start time: “01 01 1996 04 12 00”
- Simulation stop time: “02 01 1996 01 00 00”
- Time step: “0.1” minute

Remark:

- The time step of the outside domain should equal the time step of the inside domain. Because the inside domain has a five times higher resolution the resolution of the inside domain is leading in setting the time step.

6.5.2.5 Processes

The following processes need to be activated:

- Check *Sediments* and define a non-cohesive sediment fraction ‘sand’
- Check *Wind*
- Check *Wave*
- Check *Online Delft3D-WAVE*

For remarks on the wave processes see the Delft3D-WAVE user manual.

6.5.2.6 Initial conditions

Specify Uniform values as initial conditions

- Set the initial water level at “-0.45” m,
- Set the sand sediment initially to “0” kg/m^3.

6.5.2.7 Boundaries

- Press button *Open/Save*
- Import the boundary files <rif_neu.bnd>, <rif.bch> and <rif.bcc>.

DRAFT

6.5.2.8 Physical parameters

➢ Use all default values for the constants, but adapt the wind drag coefficient of the first breakpoint to “0.0025” (see Figure 6.35).

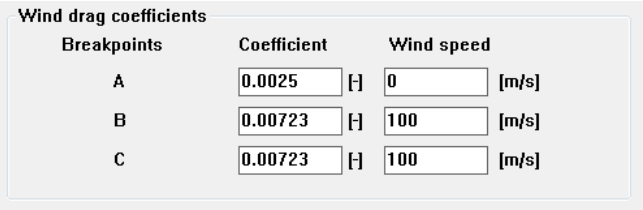

Figure 6.35: *Wind drag coefficients in Delft3D-FLOW for outside domain set-up*

➢ Set the uniform Manning roughness coefficient for both velocity components (U and V) to “0.026”.
➢ Use the default values for viscosity.
➢ Select tab *Sediment*
➢ Import file <rif_200.sed>
➢ Select tab *Morphology*
➢ Import file <rif_200.mor> files.
➢ Select tab *Wind*
➢ Import the <rif.wnd> file.

6.5.2.9 Numerical parameters

➢ Adapt the threshold depth to “0.35” m and use default values for other parameters (see Figure 6.36).

6.5.2.10 Operations

No operations are specified in this case.

6.5.2.11 Monitoring

No observation points, drogues or cross-sections are specified in this case.

6.5.2.12 Additional parameters

Add the following additional keyword (see FLOW manual Delft3D-FLOW (2013)).

➢ In column *keyword*: “Cstbnd” and
➢ in column *value*: “#Yes#”

6.5.2.13 Output

➢ Store results as specified in the figure below, see Figure 6.37.

➢ Save the file as <rif_outside.mdf>,

Exit the FLOW-GUI:

➢ Click *File* → *Exit.*

6.5.2.14 Model set-up inside FLOW domain

DRAFT

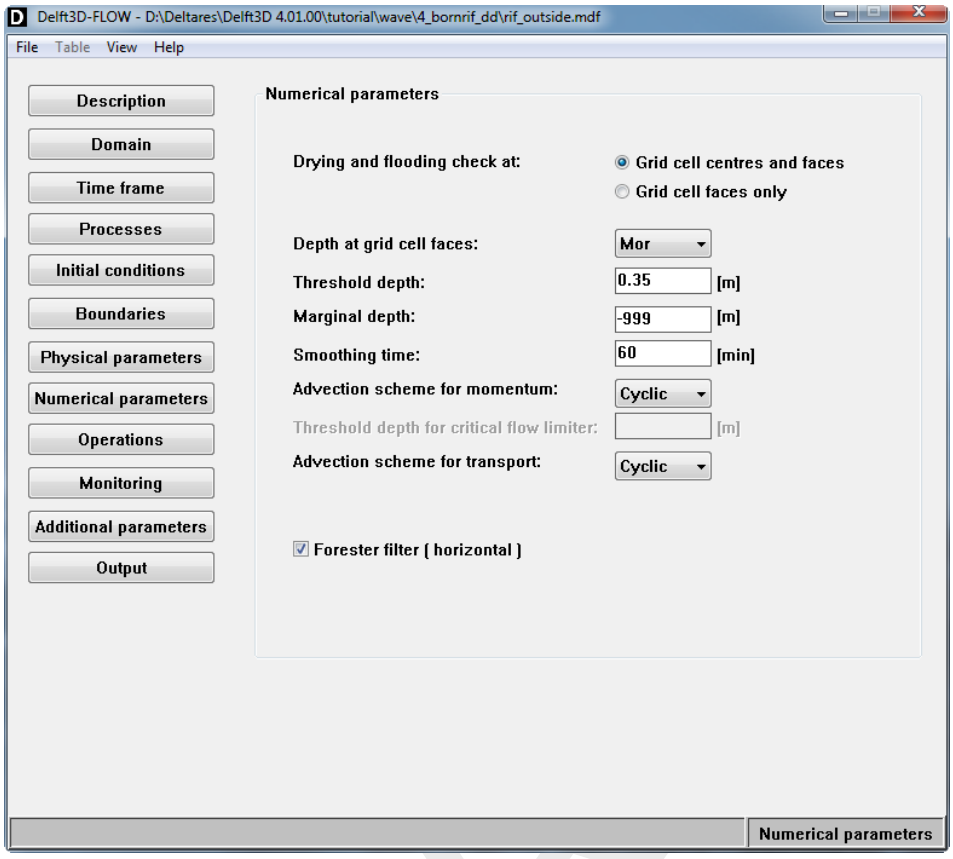

Figure 6.36: *Numerical parameters in Delft3D-FLOW for outside domain setup*

6.5.2.15 Description

Type the description:

➢ "Tutorial Delft3D-WAVE"
"Ameland Tidal Inlet"
"Combining FLOW-DD and WAVE"
"Inside model"

6.5.2.16 Domain

➢ Select tab *Grid*
➢ Import grid file <rif_inside.grd> and grid enclosure file <rif_inside.enc>
➢ Set the latitude to "52" degrees.
➢ Select tab *Bathymetry*
➢ Import the corresponding bathymetry file <rif_inside.dep>.

Dry points and thin dams are not specified in this case

Set the Data Groups *Time frame*, *Processes* and *Initial conditions* the same as the 'Outside domain FLOW model'.

DRAFT

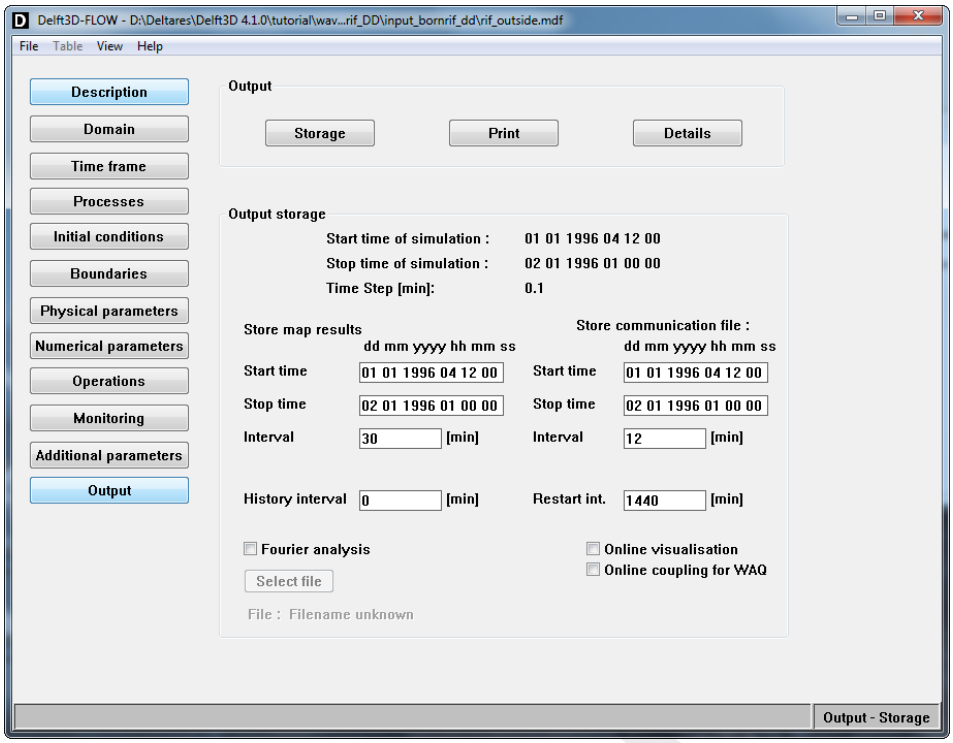

Figure 6.37: *Overview of output parameters of the Delft3D-FLOW model for the outside domain*

6.5.2.17 Boundaries

Do not specify any boundary and boundary condition! Boundary conditions are coming from the 'Outside domain FLOW model'.

Set the Data Groups *Physical parameters*, *Numerical parameters*, *Operations*, *Monitoring*, *Additional parameters* and *Output* the same as the 'Outside domain FLOW model', see section 6.5.2.1.

➢ Save the file as <rif_inside.mdf>.

Exit the FLOW-GUI:

➢ Click *File* → *Exit.*

6.5.3 Delft3D-WAVE model

6.5.3.1 Description

Type the description:

➢ Project name: "Bornrif"
➢ Project: "004"
➢ Description: "Tutorial Delft3D-WAVE"
"FLOW-DD with Online WAVE"

DRAFT

6.5.3.2 Hydrodynamics

Select the hydrodynamic results from Delft3D-FLOW

- Check checkbox *Run WAVE together with FLOW*
- Select FLOW file <rif_outside.mdf>.

Remark:

- Any mdf-file from the sub-domains can be selected. During the simulation Delft3D-WAVE will search for all mdf-files in case of a DD-simulation.

6.5.3.3 Grids

In this exercise, the detailed grid is nested in an overall grid.

Computational grid

- Import the overall grid file: <wave_overall.grd>.
- Select tab *Bathymetry*.
- Import the related bathymetry file <wave_overall.dep>.
- Import the detailed grid file: <wave_detail.grd>.
- Import the related bathymetry file <wave_detail.dep>.

This grid must be nested in the overall grid. Use for both grids the spectral resolution as default values.

Hydrodynamics

- Select tab *Hydrodynamics*.

Select for both grids the items as follows:

- **Water level** option “Use but don’t extend”.
- **Current** option “Use but don’t extend”.
- **Bathymetry** option “Use but don’t extend”.
- **Wind** option “Don’t use”.

6.5.3.4 Time frame

Communication between FLOW and WAVE will not be determined by the imported time points, but by the communication time settings specified in the FLOW-file.

6.5.3.5 Boundaries

Create the three boundaries North, East and West. First the North boundary, see Figure 6.38.

- Press the *Add*
- Set *Boundary name* to “Boundary North”
- Set *Define boundary by* to “Orientation”
- Set *Boundary orientation* to “North”
- Set *Conditions along boundary* to “Uniform”
- Set *Specification of spectra* to “Parametric”
- Press button *Edit conditions*

- Significant wave height: “2.0” m
- Peak period T_p: “7.0” s
- Direction (nautical): “330” degrees

DRAFT

➢ Directional spreading: “4” [-]

The boundaries East and West use the same values, accept for the word ’North’.

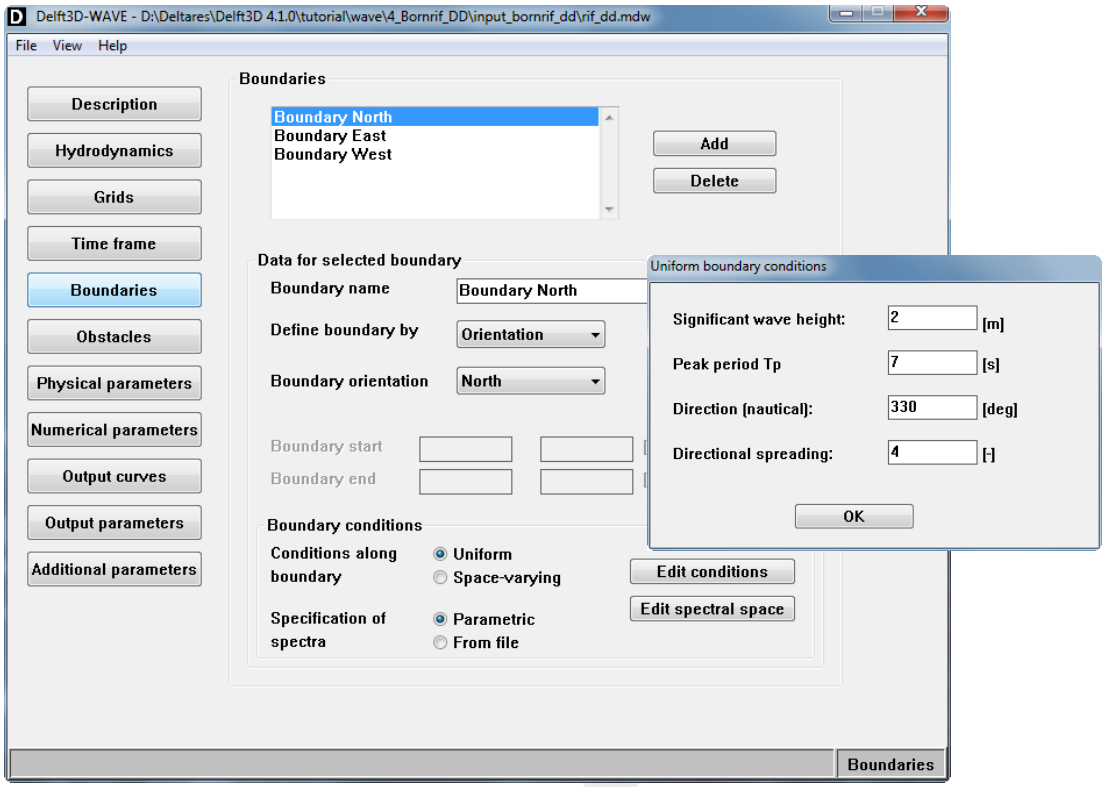

Figure 6.38: *Wave boundary conditions for ‘Boundary North’ in the WAVE model set-up*

6.5.3.6 Obstacles

No obstacles are defined

6.5.3.7 Physical parameters

➢ Press button *Wind*
➢ Specify a uniform wind speed of “6” m/s and
➢ specify a wind direction of “330” degrees.

Use default settings for all other processes.

6.5.3.8 Numerical parameters

Keep the default values (see Figure 6.39)

6.5.3.9 Output curves

No output curves are specified

6.5.3.10 Output parameters

➢ Set the output parameters as specified in Figure 6.40.

The name of the mdw-file does not have to be the same as the name of the coupled mdf-file. During the simulation Delft3D-WAVE will search for all available mdf-files in the directory.

➢ Save the wave input file as <rif_dd.mdw>.

Exit the WAVE-GUI:

➢ Click *File* → *Exit.*

6.5.4 Run and postprocessing

6.5.4.1 Foreground

To start in foreground

➢ Select *Start DD* in the **Hydrodynamics (including morphology)** menu.
➢ Select the FLOW input file <inside_outside.ddb>.
➢ Confirm the selection by pressing *OK*.
➢ Select the WAVE input file <rif_dd.mdw>.
➢ Confirm by *OK* and the flow-wave computation will be carried out.

6.5.4.2 Background

➢ Go back to the main Delft3D menu and click *Batch* (see Figure 6.33).
➢ Click *Prepare DD*.
➢ Click *Start*.

Or the complete simulation can be started with the following batch-file:

```
@ echo off
rem ============ set exedir for FLOW executable
set exedirflow=d:\delft3d\w32\flow\bin

rem ============ set exedir for WAVE executable
set exedirwave=d:\delft3d\w32\wave\bin

rem ============ set ddb-file
set ddb-file=inside_outside.ddb
echo -c %ddb-file% > delftflow.inp

rem ============ remove old output files
del runid
del TMP*.*
del trid*.*
del trih*.*
del trim*.*
del *.msg
del com*.*
del fourier*.*
del md-diag*.*

echo === start wave.exe ===
start %exedirwave%\wave.exe rif_dd.mdw 1

echo === start delftflow.exe ===
%exedirflow%\delftflow.exe delftflow.inp delftflow.out delft3d
```

DRAFT

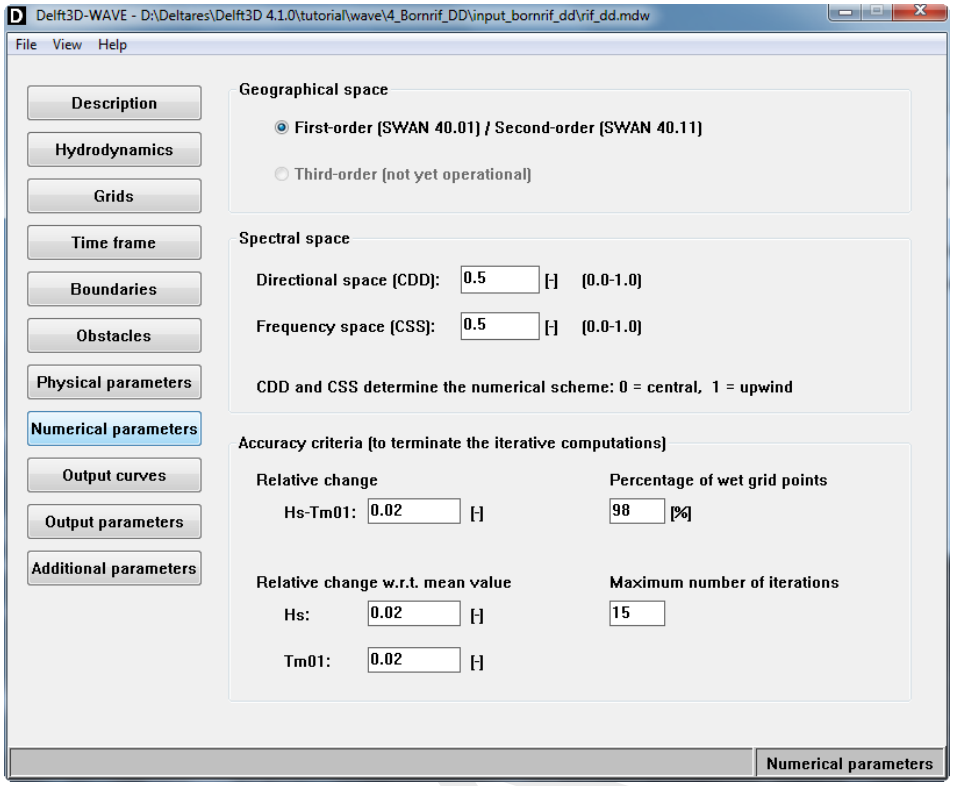

Figure 6.39: *Numerical parameters used in the WAVE model set-up*

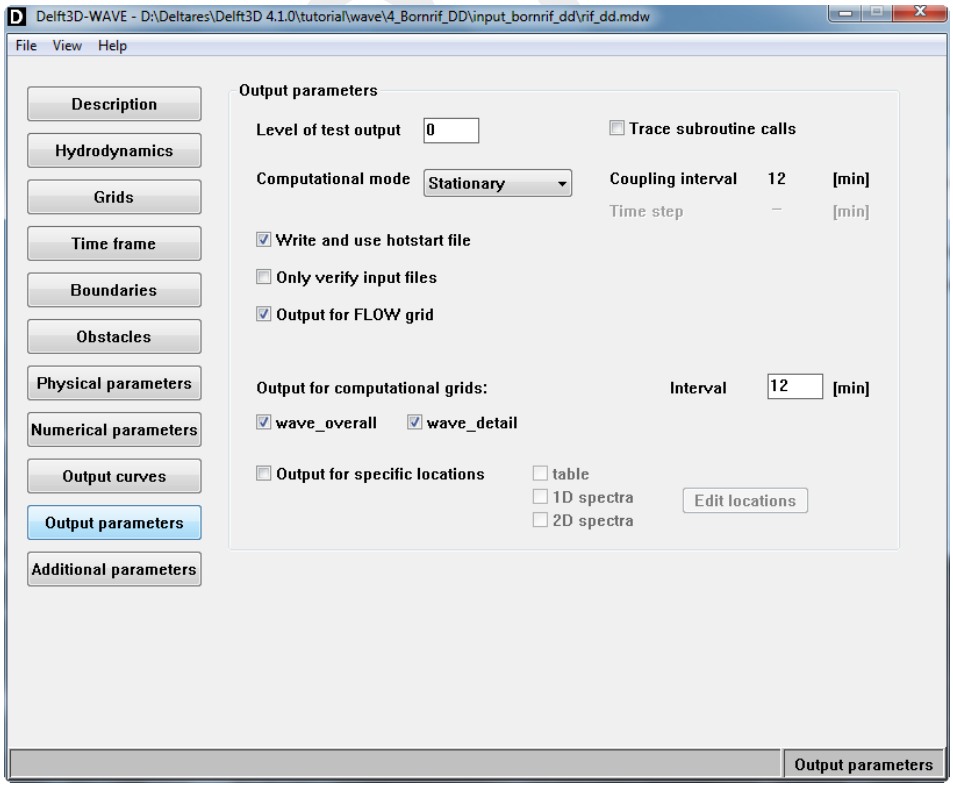

Figure 6.40: *Overview of out parameters in Delft3D WAVE.*

```
echo === end of the simulation ===

rem ============ break before deleting output and closing window
pause

rem ============ remove new output files
rem del runid
rem del TMP*.*
rem del *.msg
rem del com*.*
rem del fourier*.*
rem del md-diag*.*
```

6.5.4.3 Output files

The simulation results are stored in:

- ◇ <trim-rif_inside.dat> and <trim-rif_inside.def>
- ◇ <trim-rif_outside.dat> and <trim-rif_outside.def>
- ◇ <wavm-rif_dd-wave_detail.dat> and <wavm-rif_dd-wave_detail.def>
- ◇ <wavm-rif_dd-wave_overall.dat> and <wavm-rif_dd-wave_overall.def>

With the use of MATLAB or Delft3D-QUICKPLOT, the results can be visualised.

DRAFT

DRAFT

7 Conceptual description

7.1 Introduction

The purpose of this chapter is to give some general background with respect to the unit and co-ordinate system, the grids (resolution, orientation etc.) and the boundary conditions of the SWAN model. Advice will be given how to choose the basic input for Delft3D-WAVE for the SWAN computations.

A brief description is given with respect to the physics (see section 7.3) and numerics (section 7.4) that have been implemented in the SWAN model. This description has been copied - with permission of Delft University of Technology, The Netherlands (personal communication with dr N. Booij and dr L.H. Holthuijsen, 1999) - from the SWAN manual for SWAN version 40.41. The description given here is indicative only. For a full and proper description reference is made to SWAN (2000).

7.2 General background

7.2.1 Units and co-ordinate systems

Delft3D-WAVE expects all quantities that are input by the user, to be expressed by means of the S.I. system of units: m, kg, s and composites of these with accepted compounds, such as Newton [N] and Watt [W]. Consequently the wave height and water depth are in [m], wave period in [s] etc. Directions and spherical co-ordinates are in degrees [°] and not in radians. Delft3D-WAVE can operate in a flat plane and on a spherical earth.

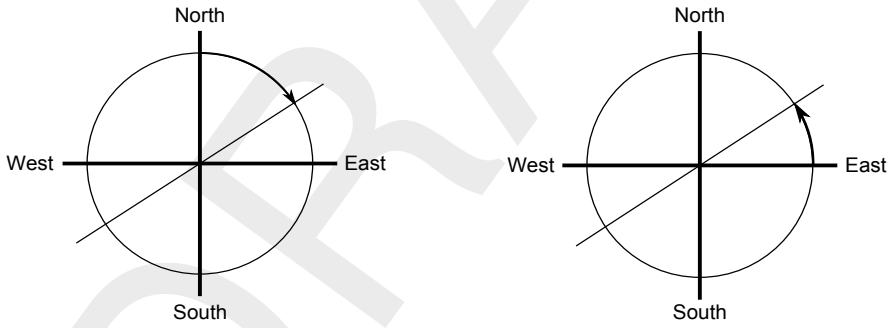

Figure 7.1: *Nautical convention (left panel) and Cartesian convention (right panel) for direction of winds and (incident) waves*

In the input for Delft3D-WAVE the directions of winds and (incident) waves are defined relative to the co-ordinate system according to a Nautical convention or Cartesian convention, see Figure 7.1 (for definitions reference is made to Appendix B).

In the Cartesian system, all geographic locations and orientations in SWAN, e.g. for the computational grid or for output points, are defined in one common Cartesian co-ordinate system with origin (0,0) by definition. This geographical origin may be chosen totally arbitrarily by you.

In the spherical system, all geographic locations and orientations in Delft3D-WAVE are defined in geographic longitude and latitude. Both co-ordinate systems are designated in this manual as the problem co-ordinate system. Figure 7.2 shows how the locations of the various grids are determined with respect to the problem co-ordinates.

DRAFT

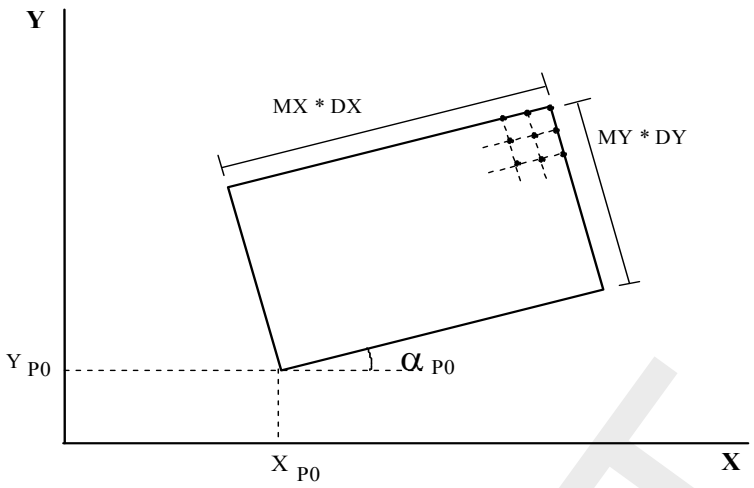

Figure 7.2: *Definition of grids (input, computational and output grids) in Delft3D-WAVE*

7.2.2 Choice of grids and boundary conditions

For your convenience Delft3D-WAVE accepts input and provides output on different grids.

It is not uncommon that a bottom grid is available as an existing data set without any relation whatsoever to Delft3D-WAVE. You may want output on an entirely different grid (but in the same region of course), whereas the computations in Delft3D-WAVE may require a different grid altogether.

For these reasons Delft3D-WAVE operates with different grids (each may have a different origin, orientation and resolution).

Input grids on which the bathymetry, current field and wind field (if present) are given by you; one computational grid on which Delft3D-WAVE performs the computations, and one (or more) output grid(s) on which you require output of Delft3D-WAVE.

During the computations (on the computational grid) Delft3D-WAVE obtains bathymetry and current information by bilinear interpolation from the input grid. The output on the output grid is in turn obtained in Delft3D-WAVE by interpolation from the computational grid. These interpolations will cause some loss of accuracy.

Input grids

Bathymetry and current input need to be provided to Delft3D-WAVE on so-called input grids (they need not be identical with the computational, the output grids or other input grids). It is best to make an input grid larger than the computational grid, in fact, so large that it completely covers the computational grid for every expected situation. In the region outside the input grid Delft3D-WAVE assumes that the bottom level and friction coefficient are identical to those at the nearest boundary of the input grid (lateral shift from that boundary). In the regions not covered by this lateral shift (i.e. in the outside corner quadrants of the input grid), a constant field equal to the value at the nearest corner point of the input grid is taken.

You should choose the resolution for the input grid such that relevant spatial details in the bathymetry and in the current pattern are well resolved. Special care is required in cases with sharp and shallow ridges in the sea bottom. In such cases the shallowest parts are of vital importance to obtain good Delft3D-WAVE results (during propagation the waves are ‘clipped’ by surf breaking at some maximum value determined by the minimum depth). To represent

DRAFT

these shallowest parts in the bottom grid, you may want to have one grid line coincide with the ridge top (even if this means ”moving” the ridge to the nearest line in the bathymetry grid). If this is not done, the computed wave height behind the shoal may well be computed higher than it is in reality, because the ridge is seen deeper in Delft3D-WAVE than it actually is (too coarse resolution to see shallow peak of the ridge).

Computational grid and boundary conditions

The computational grid is a grid in four dimensions: x-, y- and θ-, σ- space. The computational grid in x-, y-space must be chosen by you with care. You should choose the location of the up-wave boundary in water so deep that refraction effects have not (yet) influenced the wave field. However, a deep water up-wave boundary is not a strict requirement for Delft3D-WAVE. This advice is not applicable if the incoming waves are provided by a model which takes refraction into account, for instance Delft3D-WAVE itself (in a nested mode).

The computational grid must be larger than the area where you want to know the wave parameters. The length (in x-direction) needs not be longer than from the up-wave boundary to the most down-wave point of interest. The width (in y-direction) must be larger than that of the area of interest, because along each lateral side of the grid (if there is an open boundary along that side) a region exists where the wave field is disturbed (in Delft3D-WAVE) by an import of zero energy from the lateral boundaries (see Figure 7.3). This is not the case if the wave conditions along the lateral boundaries are specified by you or obtained from a previous Delft3D-WAVE run or if that boundary is closed (e.g. by land). The angle of the line dividing the disturbed area from the undisturbed area from the up-wave corner points (of the computational grid) is approximately equal to the half-power width of the directional energy distribution of the waves (this half-power width is typically 20° to 40° for waves generated by the local wind or 5° to 10° for swell).

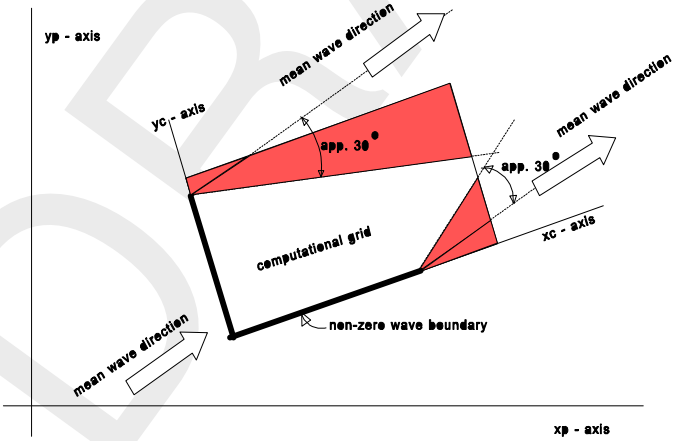

Figure 7.3: *Disturbed regions in the computational grid*

The spatial resolution of the computational grid should be sufficient to resolve relevant details of the wave field. Usually a good choice is to take the resolution of the computational grid approximately equal to that of the input (bathymetry/current) grid.

The computational spectral grid needs also to be provided by you. In frequency space it is simply defined by a minimum and maximum frequency and the frequency resolution which is proportional to the frequency itself (e.g. $\Delta f = 0.1f$). In the frequency domain this lowest frequency and highest frequency and the number of frequencies must be chosen. The value of lowest frequency must be slightly smaller than 0.6 times the value of the lowest peak frequency expected. The value of the highest frequency must be at least 2.5 to 3 times the highest peak frequency expected; usually it is chosen less than or equal to 1 Hz.

DRAFT

In directional space the directional range is the full 360° unless you specify a limited directional range. This may be convenient (less computer time and/or space) when waves travel towards a coast within a limited sector of 180°, say. The directional resolution is determined by the number of discrete directions that is provided by you. For wind seas with a directional spreading of typically 30° on either side of the mean wave direction, a resolution of 10° seems enough whereas for swell with a directional spreading of less than 10°, a resolution of 2° or less may be required. If you are confident that no energy will occur outside a certain directional sector (or is willing to ignore this energy), then the computations by SWAN can be limited to the directional sector that does contain energy. This may often be the case of waves propagating to shore within a sector of 180° around some mean wave direction.

Nonstationary situations are simulated with the SWAN model as quasi-stationary with repeated model runs. This implies that as e.g. the flow computations progress in time, a (stationary) wave computation is performed at specified, intermediate time levels. Such stationary wave computations are usually considered to be acceptable since the travel time of the waves from the seaward boundary to the coast is mostly relatively small compared to the time scale of variations in incoming wave field, the wind or tidal induced variations in depth and currents.

7.2.3 Output grids

Delft3D-WAVE can provide output on the computational grids or on grids that are independent from the computational grid like the Delft3D-FLOW grid. It must be pointed out that the information on a flow grid is obtained from the computational grid by spatial interpolation. Therefore it is wise to choose a resolution that is fine enough to show relevant spatial details.

The spatial interpolation implies that some inaccuracies are introduced. It also implies that bathymetry or current information on an (output) plot has been obtained by interpolating twice: once from the input grid to the computational grid and once from the computational grid to the output grid. If the input, computational and output grids are identical, then no interpolation errors occur.

In the regions where the output grid does not cover the computational grid Delft3D-WAVE assumes output values equal to zero.

7.3 Physical background of SWAN

7.3.1 Action balance equation

In SWAN the waves are described with the two-dimensional wave action density spectrum, even when non-linear phenomena dominate (e.g., in the surf zone). The rational for using the spectrum in such highly non-linear conditions is that, even in such conditions it seems possible to predict with reasonable accuracy this spectral distribution of the second order moment of the waves (although it may not be sufficient to fully describe the waves statistically). The spectrum that is considered in SWAN is the action density spectrum $N(\sigma, \theta)$ rather than the energy density spectrum $E(\sigma, \theta)$ since in the presence of currents, action density is conserved whereas energy density is not (Whitham, 1974). The independent variables are the relative frequency σ (as observed in a frame of reference moving with the current velocity) and the wave direction θ (the direction normal to the wave crest of each spectral component). The action density is equal to the energy density divided by the relative frequency: $N(\sigma, \theta) = E(\sigma, \theta)/\sigma$. In SWAN this spectrum may vary in time and space.

In SWAN the evolution of the wave spectrum is described by the spectral action balance

equation which for Cartesian co-ordinates is (e.g., Hasselmann *et al.* (1973)):

$$\frac{\partial}{\partial t}N + \frac{\partial}{\partial x}c_x N + \frac{\partial}{\partial y}c_y N + \frac{\partial}{\partial \sigma}c_\sigma N + \frac{\partial}{\partial \theta}c_\theta N = \frac{S}{\sigma} \tag{7.1}$$

The first term in the left-hand side of this equation represents the local rate of change of action density in time, the second and third term represent propagation of action in geographical space (with propagation velocities c_x and c_y in x- and y-space, respectively). The fourth term represents shifting of the relative frequency due to variations in depths and currents (with propagation velocity c_σ in σ-space). The fifth term represents depth-induced and current-induced refraction (with propagation velocity c_θ in θ-space). The expressions for these propagation speeds are taken from linear wave theory (Whitham, 1974; Mei, 1983; Dingemans, 1997). The term $S\ (= S(\sigma, \theta))$ at the right-hand side of the action balance equation is the source term in terms of energy density representing the effects of generation, dissipation and non-linear wave-wave interactions. A brief summary of the formulations that are used for the various source terms in SWAN is given next.

The following processes are accounted for in SWAN:

- ◇ generation by wind,
- ◇ dissipation by whitecapping, bottom friction and depth-induced breaking,
- ◇ non-linear wave-wave interaction (quadruplets and triads).

In addition wave propagation through obstacles and wave-induced set-up of the mean sea surface can be computed in SWAN. These phenomena are addressed separately below (see Sections 7.3.2 and 7.3.3).

Wind input

Transfer of wind energy to the waves is described in SWAN with a resonance mechanism (Phillips, 1957) and a feed-back mechanism (Miles, 1957). The corresponding source term for these mechanisms is commonly described as the sum of linear and exponential growth:

$$S_{in}(\sigma, \theta) = A + BE(\sigma, \theta) \tag{7.2}$$

in which A and B depend on wave frequency and direction, and wind speed and direction. The effects of currents are accounted for in SWAN by using the apparent local wind speed and direction. The expression for the term A is due to Cavaleri and Malanotte-Rizzoli (1981) with a filter to avoid growth at frequencies lower than the Pierson-Moskowitz frequency (Tolman, 1992a). Two optional expressions for the coefficient B are used in the model. The first is taken from an early version of the WAM model (known as WAM Cycle 3, the WAMDI group (1988)). It is due to Snyder *et al.* (1981), rescaled in terms of friction velocity U_* by Komen *et al.* (1984). The drag coefficient to relate U_* to the driving wind speed at 10 m elevation U_{10} is taken from Wu (1982). The second expression for B in SWAN is taken from the most recent version of the WAM model (known as WAM Cycle 4, Komen *et al.* (1994)). It is due to Janssen (1991a) and it accounts explicitly for the interaction between the wind and the waves by considering atmospheric boundary layer effects and the roughness length of the sea surface. The corresponding set of equations is solved (as in the WAM model) with the iterative procedure of Mastenbroek *et al.* (1993).

DRAFT

Dissipation

The dissipation term of wave energy is represented by the summation of three different contributions: whitecapping $S_{ds,w}(\sigma,\theta)$, bottom friction $S_{ds,b}(\sigma,\theta)$ and depth-induced breaking $S_{ds,br}(\sigma,\theta)$.

Whitecapping is primarily controlled by the steepness of the waves. In presently operating third-generation wave models (including SWAN) the whitecapping formulations are based on a pulse-based model (Hasselmann, 1974), as adapted by the WAMDI group (1988):

$$S_{ds,w}(\sigma,\theta) = -\Gamma\tilde{\sigma}\frac{k}{\tilde{k}}E(\sigma,\theta) \tag{7.3}$$

where Γ is a steepness dependent coefficient, k is the wave number and $\tilde{sigma}$ and $\tilde{k}$ denote a mean frequency and a mean wave number, respectively (cf. the WAMDI group (1988)). Komen *et al.* (1984) estimated the value of Γ by closing the energy balance of the waves in fully developed conditions. This implies that this value depends on the wind input formulation that is used.

An alternative description for whitecapping in SWAN is given by Van der Westhuysen *et al.* (2007) and Van der Westhuysen (2007), which is an adapted form of the expression of Alves and Banner (2003). The latter is based on the apparent relationship between wave groups and whitecapping dissipation. This adaption is due to the fact that it can also be applied to mixed sea-swell conditions and in shallow water. This was done by removing the dependencies on mean spectral steepness and wavenumber in the original expression, and by applying source term scaling arguments for its calibration (see below). This led to the following expression for whitecapping dissipation:

$$S_{ds,w}(\sigma,\theta) = -C'_{ds}\left(\frac{B(k)}{B_r}\right)^{p/2}(\tanh(kh))^{(2-p_0)/4}\sqrt{gk}E(\sigma,\theta) \tag{7.4}$$

in which the density function B(k) is the azimuthal-integrated spectral saturation, which is positively correlated with the probability of wave group-induced breaking. It is calculated from frequency space variables as follows:

$$B(k) = \int_0^{2\pi} c_g k^3 E(\sigma,\theta)d\theta \tag{7.5}$$

and $B_r = 1.75\times10^{-3}$ is a threshold saturation level. The proportionality coefficient is set to $C'_{ds} = 5.0\times10^{-5}$. When $B(k) > B_r$, waves break and the exponent p is set equal to a calibration parameter p_0. For $B(k) \leq B_r$ there is no breaking, but some residual dissipation proved necessary. This is obtained by setting $p = 0$.

Depth-induced dissipation may be caused by bottom friction, by bottom motion, by percolation or by back-scattering on bottom irregularities (Shemdin *et al.*, 1978). For continental shelf seas with sandy bottoms, the dominant mechanism appears to be bottom friction (e.g., Bertotti and Cavaleri (1994)) which can generally represented as:

$$S_{ds,b}(\sigma,\theta) = -C_{bottom}\frac{\sigma^2}{g^2\sinh^2(kd)}E(\sigma,\theta) \tag{7.6}$$

in which C_{bottom} is a bottom friction coefficient. A large number of models have been proposed since the pioneering paper of Putnam and Johnson (1949). Hasselmann *et al.* (1973)

DRAFT

suggested to use an empirically obtained constant. It seems to perform well in many different conditions as long as a suitable value is chosen (typically different for swell and wind sea; Bouws and Komen (1983)). A non-linear formulation based on drag has been proposed by Hasselmann and Collins (1968) which was later simplified by Collins (1972). More complicated, eddy viscosity models have been developed by Madsen *et al.* (1988) (see Weber (1991a)) and by Weber (1989, 1991a,b). Considering the large variations in bottom conditions in coastal areas (bottom material, bottom roughness length, ripple height etc.), there is no field data evidence to give preference to a particular friction model (Luo and Monbaliu, 1994). For this reason, the simplest of each of these types of friction models has been implemented in SWAN: the empirical JONSWAP model of Hasselmann *et al.* (1973), the drag law model of Collins (1972) and the eddy-viscosity model of Madsen *et al.* (1988). The effect of a mean current on the wave energy dissipation due to bottom friction is not taken into account in SWAN. The reasons for this are given by Tolman (1992b) who argues that state-of-the-art expressions vary too widely in their effects to be acceptable. He found that the error in finding a correct estimate of the bottom roughness length scale has a much larger impact on the energy dissipation rate than the effect of a mean current.

The process of depth-induced wave-breaking is still poorly understood and little is known about its spectral modelling. In contrast to this, the total dissipation (i.e., integrated over the spectrum) due to this type of wave breaking can be well modelled with the dissipation of a bore applied to the breaking waves in a random field (Battjes and Janssen, 1978; Thornton and Guza, 1983). Laboratory observations (e.g., Battjes and Beji (1992), Vincent *et al.* (1994); Arcilla *et al.* (1994) and Eldeberky and Battjes (1996)) show that the shape of initially unimodal spectra propagating across simple (barred) beach profiles, is fairly insensitive to depth-induced breaking. This has led Eldeberky and Battjes (1995) to formulate a spectral version of the bore model of Battjes and Janssen (1978) which conserves the spectral shape. Expanding their expression to include directions, the expression that is used in SWAN is:

$$S_{ds,br}(\sigma,\theta) = -\frac{D_{tot}}{E_{tot}} E(\sigma,\theta) \tag{7.7}$$

in which E_{tot} and D_{tot} is the rate of dissipation of the total energy due to wave breaking according to Battjes and Janssen (1978). Adding a quadratic dependency on frequency as suggested by Mase and Kirby (1992) (supported by Elgar *et al.* (1997)) seems to have no noticeable effect on the SWAN results. Chen and Guza (1997) inferred from observations and simulations with a Boussinesq model that the high-frequency levels are insensitive to such frequency dependency because an increased dissipation at high frequencies is compensated approximately by increased non-linear energy transfer (but they did find the frequency dependency to be relevant in time domain). The value of D_{tot} depends critically on the breaking parameter $\gamma = H_{max}/d$ (in which H_{max} is the maximum possible individual wave height in the local water depth d). In Delft3D-WAVE a constant value is available equal to $\gamma = 0.73$ (the mean value of the data set of Battjes and Stive (1985).

Non-linear wave-wave interactions

In deep water, quadruplet wave-wave interactions dominate the evolution of the spectrum. They transfer wave energy from the spectral peak to lower frequencies (thus moving the peak frequency to lower values) and to higher frequencies (where the energy is dissipated by white-capping). In very shallow water, triad wave-wave interactions transfer energy from lower frequencies to higher frequencies often resulting in higher harmonics (Beji and Battjes, 1993) (low-frequency energy generation by triad wave-wave interactions is not considered here).

A full computation of the quadruplet wave-wave interactions is extremely time consuming and not convenient in any operational wave model. A number of techniques, based on parametric methods or other types of approximations have been proposed to improve computational

speed (see Young and Van Vledder (1993) for a review). In SWAN the computations are carried out with the Discrete Interaction Approximation (DIA) of Hasselmann *et al.* (1985). This DIA has been found quite successful in describing the essential features of a developing wave spectrum (Komen *et al.*, 1994). For uni-directional waves, this approximation is not valid. In fact, the quadruplet interaction coefficient for these waves is nearly zero (G.Ph. van Vledder, personal communication, 1996). For finite-depth applications, Hasselmann and Hasselmann (1981) have shown that for a JONSWAP-type spectrum the quadruplet wave-wave interactions can be scaled with a simple expression (it is used in SWAN).

A first attempt to describe triad wave-wave interactions in terms of a spectral energy source term was made by Abreu *et al.* (1992). However, their expression is restricted to non-dispersive shallow water waves and is therefore not suitable in many practical applications of wind waves. The breakthrough in the development came with the work of Eldeberky and Battjes (1995) who transformed the amplitude part of the Boussinesq model of Madsen and Sørensen (1993) into an energy density formulation and who parameterised the biphase of the waves on the basis of laboratory observations (Battjes and Beji, 1992; Arcilla, Roelvink, O'Connor, Reniers and Jimenez, 1994). A discrete triad approximation (DTA) for co-linear waves was subsequently obtained by considering only the dominant self-self interactions. Their model has been verified with flume observations of long-crested, random waves breaking over a submerged bar (Beji and Battjes, 1993) and over a barred beach (Arcilla *et al.*, 1994). The model appeared to be fairly successful in describing the essential features of the energy transfer from the primary peak of the spectrum to the super harmonics. A slightly different version, the Lumped Triad Approximation (LTA) was later derived by Eldeberky and Battjes (1996). This LTA is used in SWAN.

7.3.2 Propagation through obstacles

SWAN can estimate wave transmission through a (line-)structure such as a breakwater (dam). Such an obstacle will affect the wave field in two ways, first it will reduce the wave height locally all along its length, and second it will cause diffraction around its end(s). The model is not able to account for diffraction. In irregular, short-crested wave fields, however, it seems that the effect of diffraction is small, except in a region less than one or two wavelengths away from the tip of the obstacle (Booij *et al.*, 1992). Therefore the model can reasonably account for waves around an obstacle if the directional spectrum of incoming waves is not too narrow. Since obstacles usually have a transversal area that is too small to be resolved by the bathymetry grid in SWAN, an obstacle is modelled as a line. If the crest of the breakwater is at a level where (at least part of the) waves can pass over, the transmission coefficient K_t (defined as the ratio of the (significant) wave height at the down-wave side of the dam over the (significant) wave height at the up-wave side) is a function of wave height and the difference in crest level and water level. The expression is taken from Goda *et al.* (1967):

$$K_t = 0.5\left[1 - \sin\left(\frac{\pi}{2\alpha}\left(\frac{F}{H_i} + \beta\right)\right)\right] \qquad \text{for} \qquad -\beta - \alpha < \frac{F}{H_i} < \alpha - \beta \quad (7.8)$$

where $F = h - d$ is the freeboard of the dam and where H_i is the incident (significant) wave height at the up-wave side of the obstacle (dam), h is the crest level of the dam above the reference level (same as reference level of the bottom), d the mean water level relative to the reference level, and the coefficients α, β depend on the shape of the dam (Seelig, 1979):

Case	α	β
Vertical thin wall	1.8	0.1
Caisson	2.2	0.4
Dam with slope 1:3/2	2.6	0.15

DRAFT

The above expression is based on experiments in a wave flume, so strictly speaking it is only valid for normal incidence waves. Since there is no data available on oblique waves it is assumed that the transmission coefficient does not depend on direction. Another phenomenon that is to be expected is a change in wave frequency since often the process above the dam is highly non-linear. Again there is little information available, so in the model it is assumed that the frequencies remain unchanged over an obstacle (only the energy scale of the spectrum is affected and not the spectral shape).

7.3.3 Wave-induced set-up

In a (geographic) 1D case the computation of the wave induced set-up is based on the vertically integrated momentum balance equation which is a balance between the wave force (gradient of the wave radiation stress) and the hydrodynamic pressure gradient (no wave-induced currents exist).

$$F_x + gd\frac{\partial \bar{\eta}}{\partial x} = 0 \tag{7.9}$$

where d is the total water depth (including the wave-induced set-up) and $\bar{\eta}$ is the mean surface elevation (including the wave-induced set-up).

In a 2D case, computations are also based on the vertically integrated momentum balance equation (in two geographic dimensions), supplemented with the observation of Dingemans *et al.* (1987) that the wave-induced currents are mainly driven by the divergence-free part of the wave forces whereas the set-up is mainly due to the rotation-free part of these forces. To compute the set-up, it would then be sufficient to compute the set-up as if the currents are zero, which implies that the divergence of all forces considered would be zero:

$$\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial}{\partial x}\left(gd\frac{\partial \eta}{\partial x}\right) + \frac{\partial}{\partial y}\left(gd\frac{\partial \eta}{\partial y}\right) = 0 \tag{7.10}$$

Note that divergence = 0 is only an approximation of the true divergence. These two equations have been implemented in SWAN. The 2D set-up module can be activated within Delft3D-WAVE.

7.3.4 Diffraction

To accommodate diffraction in SWAN simulations, a phase-decoupled refraction-diffraction approximation is suggested (Holthuijsen *et al.*, 1993). It is expressed in terms of the directional turning rate of the individual wave components in the 2D wave spectrum. The approximation is based on the mild-slope equation for refraction and diffraction, omitting phase information. It does therefore not permit coherent wave fields in the computational domain.

7.4 Full expressions for source terms

The complete expressions for the physical processes of generation, dissipation and non-linear wave-wave interactions that are available in the SWAN model are given here.

7.4.1 Input by wind

Wave growth by wind is described by:

$$S_{in}(\sigma,\theta) = A + BE(\sigma,\theta) \tag{7.11}$$

DRAFT

in which A describes linear growth and BE exponential growth. It should be noted that the SWAN model is driven by the wind speed at 10 m elevation U_{10} whereas the computations use the friction velocity U_*. For the WAM Cycle 3 formulation the transformation from U_{10} to U_* is obtained with:

$$U_*^2 = C_D U_{10}^2 \tag{7.12}$$

in which C_D is the drag coefficient from Wu (1982) **?** :

$$C_D(U_{10}) = \begin{cases} 1.2875 \times 10^{-3} & \text{for} \quad U_{10} < 7.5 \text{ m/s} \\ (0.8 + 0.065\,[s/m] \times U_{10}) \times 10^{-3} & \text{for} \quad U_{10} \geq 7.5 \text{ m/s} \end{cases} \tag{7.13}$$

The expression for B is due to Komen *et al.* (1984). Their expression is a function of U_*/c_{ph}:

$$B = \max\left(0, 0.25\frac{\rho_a}{\rho_w}\left(28\frac{U_*}{c_{ph}}\cos(\theta - \theta_w) - 1\right)\right)\sigma \tag{7.14}$$

in which c_{ph} is the phase speed and ρ_a and ρ_w are the density of air and water, respectively. This expression is also used in WAM Cycle 3 (cf. the WAMDI group (1988)).

7.4.2 Dissipation of wave energy

Whitecapping

The processes of whitecapping in the SWAN model are represented by the pulse-based model of Hasselmann (1974). Reformulated in terms of wave number (rather than frequency) so as to be applicable in finite water depth (cf. the WAMDI group (1988)), this expression is:

$$S_{ds,w}(\sigma,\theta) = -\Gamma\tilde{\sigma}\frac{k}{\tilde{k}}E(\sigma,\theta) \tag{7.15}$$

where $\tilde{\sigma}$ and $\tilde{k}$ denote the mean frequency and the mean wave number (for expressions see below) respectively and the coefficient Γ depends on the overall wave steepness. This steepness dependent coefficient, as given by the WAMDI group (1988), has been adapted by Günther *et al.* (1992) based on Janssen (1991a,b):

$$\Gamma = \Gamma_{KJ} = C_{ds}\left((1-\delta) + \delta\frac{k}{\tilde{k}}\right)\left(\frac{\tilde{s}}{\tilde{s}_{PM}}\right)^p \tag{7.16}$$

For $\delta = 0$ the expression of Γ reduces to the expression as used by the WAMDI group (1988). The coefficients C_{ds}, δ and m are tunable coefficients, $\tilde{s}$ is the overall wave steepness (defined below), $\tilde{s}_{PM}$ is the value of $\tilde{s}$ for the Pierson-Moskowitz spectrum (1964; $\tilde{s}_{PM} = (3.02 \times 10^{-3})^{1/2}$). This overall wave steepness $\tilde{s}$ is defined as:

$$\tilde{s} = \tilde{k}\sqrt{E_{tot}} \tag{7.17}$$

The mean frequency $\tilde{\sigma}$, the mean wave number $\tilde{k}$ and the total wave energy E_{tot} is defined as (cf. the WAMDI group (1988)):

$$\begin{aligned} \tilde{\sigma} &= \left(E_{tot}^{-1}\int_0^{2\pi}\int_0^{\infty}\frac{1}{\sigma}E(\sigma,\theta)d\sigma d\theta\right)^{-1} \\ \tilde{k} &= \left(E_{tot}^{-1}\int_0^{2\pi}\int_0^{\infty}\frac{1}{\sqrt{k}}E(\sigma,\theta)d\sigma d\theta\right)^{-2} \\ E_{tot} &= \int_0^{2\pi}\int_0^{\infty}E(\sigma,\theta)d\sigma d\theta \end{aligned} \tag{7.18}$$

DRAFT

The values of the tunable coefficients C_{ds} and δ and exponent p in this model have been obtained by Komen *et al.* (1984) by closing the energy balance of the waves in idealised wave growth conditions (both for growing and fully developed wind seas) for deep water. This implies that coefficients in the steepness dependent coefficient Γ depend on the wind input formulation that is used. For the wind input of Komen *et al.* (1984) (corresponding to WAM Cycle 3; the WAMDI group (1988)):

$$C_{ds} = 2.36 \times 10^{-5}, \tag{7.19}$$

$$\delta = 0 \quad \text{and} \tag{7.20}$$

$$p = 4. \tag{7.21}$$

Bottom friction

The bottom friction models that have been selected for SWAN are the empirical model of JONSWAP (Hasselmann *et al.*, 1973), the drag law model of Collins (1972) and the eddy-viscosity model of Madsen *et al.* (1988). The formulations for these bottom friction models can all be expressed in the following form:

$$S_{ds,b}(\sigma,\theta) = -C_{bottom}\frac{\sigma^2}{g^2\sinh^2(kd)}E(\sigma,\theta) \tag{7.22}$$

in which C_{bottom} is a bottom friction coefficient that generally depends on the bottom orbital motion represented by U_{rms}:

$$U_{rms}^2 = \int_0^{2\pi}\int_0^{\infty}\frac{\sigma^2}{\sinh^2(kd)}E(\sigma,\theta)d\sigma d\theta \tag{7.23}$$

Hasselmann *et al.* (1973) found from the results of the JONSWAP experiment $C_{bottom} = C_{JON} = 0.038$ m^{2}s^{-3} for swell conditions. Bouws and Komen (1983) selected a bottom friction coefficient of $C_{JON} = 0.067$ m^{2}s^{-3} for fully developed wave conditions in shallow water. Both values are available in SWAN.

The expression of Collins (1972) is based on a conventional formulation for periodic waves with the appropriate parameters adapted to suit a random wave field. The dissipation rate is calculated with the conventional bottom friction formulation of Eq. 7.22 in which the bottom friction coefficient is $C_{bottom} = C_f g U_{rms}$ with $C_f = 0.015$ (Collins, 1972). (Note that Collins (1972) contains an error in the expression due to an erroneous Jacobean transformation; see page A-16 of Tolman (1990).)

Madsen *et al.* (1988) derived a formulation similar to that of Hasselmann and Collins (1968) but in their model the bottom friction factor is a function of the bottom roughness height and the actual wave conditions. Their bottom friction coefficient is given by:

$$C_{bottom} = f_w\frac{g}{\sqrt{2}}U_{rms} \tag{7.24}$$

in which f_w is a non-dimensional friction factor estimated by using the formulation of Jonsson (1966) (cf. Madsen *et al.* (1988)):

$$\frac{1}{4\sqrt{f_w}} + {}^{10}\log\left(\frac{1}{4\sqrt{f_w}}\right) = m_f + {}^{10}\log\left(\frac{a_b}{K_N}\right) \tag{7.25}$$

in which $m_f = -0.08$ (Jonsson and Carlsen, 1976) and a_b is a representative near-bottom excursion amplitude:

$$a_b^2 = 2\int_0^{2\pi}\int_0^{\infty}\frac{1}{\sinh^2(kd)}E(\sigma,\theta)d\sigma d\theta \tag{7.26}$$

DRAFT

and K_N is the bottom roughness length scale. For values of a_b/KN smaller than 1.57 the friction factor f_w is 0.30 (Jonsson, 1980).

Depth-induced wave breaking

To model the energy dissipation in random waves due to depth-induced breaking, the bore-based model of Battjes and Janssen (1978) is used in SWAN. The mean rate of energy dissipation per unit horizontal area due to wave breaking D_{tot} is expressed as:

$$D_{tot} = -\frac{1}{4}\alpha_{BJ} Q_b \left(\frac{\sigma}{2\pi}\right) H_m^2 \tag{7.27}$$

in which $\alpha_{BJ} = 1$ in SWAN, Q_b [-] is the fraction of breaking waves determined by:

$$\frac{1 - Q_b}{\ln Q_b} = -8\frac{E_{tot}}{H_m^2} \tag{7.28}$$

in which H_m is the maximum wave height that can exist at the given depth and $\bar{\sigma}$ is a mean frequency defined as:

$$\bar{\sigma} = E_{tot}^{-1} \int_0^{2\pi} \int_0^{\infty} \sigma E(\sigma, \theta) d\sigma d\theta \tag{7.29}$$

Extending the expression of Eldeberky and Battjes (1995) to include the spectral directions, the dissipation for a spectral component per unit time is calculated in SWAN with:

$$S_{ds,br}(\sigma, \theta) = D_{tot} \frac{E(\sigma, \theta)}{E_{tot}} \tag{7.30}$$

The maximum wave height H_m is determined in SWAN with $H_m = \gamma d$, in which γ is the breaker parameter and d is the total water depth (including the wave-induced set-up if computed by SWAN). In literature, this breaker parameter γ is often a constant or it is expressed as a function of bottom slope or incident wave steepness (Galvin, 1972; Battjes and Janssen, 1978; Battjes and Stive, 1985; Arcilla and Lemos, 1990; Kaminsky and Kraus, 1993; Nelson, 1987, 1994). Since SWAN is locally defined, the dependency on incident wave steepness cannot be used.

In the publication of Battjes and Janssen (1978) in which the dissipation model is described, a constant breaker parameter, based on Miche's criterion, of $\gamma = 0.8$ was used. Battjes and Stive (1985) re-analysed wave data of a number of laboratory and field experiments and found values for the breaker parameter varying between 0.6 and 0.83 for different types of bathymetry (plane, bar-trough and bar) with an average of 0.73. From a compilation of a large number of experiments Kaminsky and Kraus (1993) have found breaker parameters in the range of 0.6 to 1.59 with an average of 0.79.

7.4.3 Nonlinear wave-wave interactions

Quadruplet wave-wave interactions

The quadruplet wave-wave interactions are computed with the Discrete Interaction Approximation (DIA) as proposed by Hasselmann *et al.* (1985). Their source code (slightly adapted by Tolman, personal communication, 1993) has been used in the SWAN model. In the Discrete Interaction Approximation two quadruplets of wave numbers are considered, both with

DRAFT

frequencies:

$$\begin{aligned}\sigma_1 &= \sigma_2 = \sigma \\ \sigma_3 &= \sigma(1+\lambda) = \sigma^+ \\ \sigma_4 &= \sigma(1-\lambda) = \sigma^-\end{aligned} \tag{7.31}$$

where λ is a constant coefficient set equal to 0.25. To satisfy the resonance conditions for the first quadruplet, the wave number vectors with frequency σ_3 and σ_4 lie at an angle of $\theta_1 = -11.5°$ and $\theta_2 = 33.6°$ to the two identical wave number vectors with frequencies σ_1 and σ_2. The second quadruplet is the mirror of this first quadruplet (the wave number vectors with frequency σ_3 and σ_4 lie at mirror angles of $\theta_3 = 11.5°$ and $\theta_4 = -33.6°$.

Within this discrete interaction approximation, the source term $S_{nl4}(\sigma,\theta)$ is given by:

$$S_{nl4}(\sigma,\theta) = S^*_{nl4}(\sigma,\theta) + S^{**}_{nl4}(\sigma,\theta) \tag{7.32}$$

where $S^*_{nl4}(\sigma,\theta)$ refers to the first quadruplet and $S^{**}_{nl4}(\sigma,\theta)$ to the second quadruplet (the expressions for $S^{**}_{nl4}(\sigma,\theta)$ are identical to those for $S^*_{nl4}(\sigma,\theta)$ for the mirror directions) and:

$$S^*_{nl4}(\sigma,\theta) = 2\delta S^*_{nl4}(\alpha_1,\sigma,\theta) - \delta S^*_{nl4}(\alpha_2,\sigma,\theta) - \delta S^*_{nl4}(\alpha_3,\sigma,\theta) \tag{7.33}$$

in which $\alpha_1 = 1$, $\alpha_2 = (1+\lambda)$ and $\alpha_3 = (1-\lambda)$. Each of the contributions ($i = 1,2,3$) is:

$$\begin{aligned}&\delta S_{nl4}(\alpha_i\sigma,\theta) = C_{nl4}(2\pi)^2 g^{-4}\left(\frac{\sigma}{2\pi}\right)^{11} \\ &\left[E^2(\alpha_i\sigma,\theta)\left(\frac{E^2(\alpha_i\sigma^+,\theta)}{(1+\lambda)^4} + \frac{E^2(\alpha_i\sigma^-,\theta)}{(1-\lambda)^4}\right) - 2\frac{E^2(\alpha_i\sigma,\theta)E^2(\alpha_i\sigma^+,\theta)E^2(\alpha_i\sigma^-,\theta)}{(1-\lambda^2)^4}\right]\end{aligned} \tag{7.34}$$

The constant $C_{nl4} = 3\times10^7$. Following Hasselmann and Hasselmann (1981), the quadruplet interaction in finite water depth is taken identical to the quadruplet transfer in deep water multiplied with a scaling factor R:

$$S_{nl4,\textit{finite depth}} = R(k_p d) S_{nl4,\textit{infinite depth}} \tag{7.35}$$

where R is given by:

$$R(k_p d) = 1 + \frac{C_{sh1}}{k_p d}(1 - C_{sh2} k_p d)\exp(C_{sh3} k_p d) \tag{7.36}$$

in which k_p is the peak wave number of the JONSWAP spectrum for which the original computations were carried out. The values of the coefficients are: $C_{sh1} = 5.5$, $C_{sh2} = 6/7$ and $C_{sh3} = -1.25$. In the shallow water limit, i.e., $k_p d \to 0$ the non-linear transfer tends to infinity. Therefore a lower limit of $k_p d = 0.5$ is applied (cf. WAM Cycle 4; Komen *et al.* (1994), resulting in a maximum value of $R(k_p d) = 4.43$. To increase the model robustness in case of arbitrarily shaped spectra, the peak wave number k_p is replaced by $k_p = 0.75\bar{k}$ (Komen *et al.*, 1994).

DRAFT

Triad wave-wave interactions

The Lumped Triad Approximation (LTA) of Eldeberky and Battjes (1996), which is a slightly adapted version of the Discrete Triad Approximation of Eldeberky and Battjes (1995) is used in SWAN in each spectral direction:

$$S_{nl3}(\sigma,\theta) = S^{-}_{nl3}(\sigma,\theta) + S^{+}_{nl3}(\sigma,\theta) \tag{7.37}$$

with

$$S^{+}_{nl3}(\sigma,\theta) = \max\{0, \alpha_{EB} 2\pi c c_g J^2 |\sin(\beta)| \{E^2(\sigma/2,\theta) - 2E(\sigma/2,\theta)E(\sigma,\theta)\}\} \tag{7.38}$$

and

$$S^{-}_{nl3}(\sigma,\theta) = -2S^{+}_{nl3}(2\sigma,\theta) \tag{7.39}$$

in which α_{EB} is a tunable proportionality coefficient. The bi-phase β is approximated with

$$\beta = -\frac{\pi}{2} + \frac{\pi}{2}\tanh\left(\frac{0.2}{Ur}\right) \tag{7.40}$$

with Ursell number Ur:

$$Ur = \frac{g}{8\sqrt{2}\pi^2}\frac{H_s \bar{T}^2}{d^2} \tag{7.41}$$

with $\bar{T} = 2\pi/\bar{\sigma}$. Usually, the triad wave-wave interactions are calculated only for $0.1 \leq Ur \leq 10$. But for stability reasons, it is calculated for the whole range $0 \leq Ur \leq 10$. This means that both quadruplets and triads are computed at the same time. The interaction coefficient J is taken from Madsen and Sørensen (1993):

$$J = \frac{k^2_{\sigma/2}(gd + 2c^2_{\sigma/2})}{k_\sigma d\left(gd + \frac{2}{15}gd^3k^2_\sigma - \frac{2}{5}\sigma^2 d^2\right)} \tag{7.42}$$

Wave-induced set-up

In a geographic 1D case the computation of the wave induced set-up is based on the vertically integrated momentum balance equation which is a balance between the wave force (gradient of the wave radiation stress normal to the coast) and the hydrostatic pressure gradient (note that the component parallel to the coast causes wave-induced currents but no set-up):

$$\frac{dS_{xx}}{dx} + \rho g H \frac{d\bar{\eta}}{dx} = 0 \tag{7.43}$$

where $H = d + \bar{\eta}$ is the total water depth (including the wave-induced set-up), d is the bottom level, $\bar{\eta}$ is the mean surface elevation (including the wave-induced set-up) and

$$S_{xx} = \rho g \int \left(n\cos^2\theta + \frac{n-1}{2}\right) E \, d\sigma d\theta \tag{7.44}$$

is the radiation stress tensor.

Observation and computations based on the vertically integrated momentum balance equation of Dingemans *et al.* (1987) show that the wave-induced currents are mainly driven by the divergence-free part of the wave forces whereas the set-up is mainly due to the rotation-free part of these forces. To compute the set-up, it would then be sufficient to consider the divergence of the momentum balance equation. If the divergence of the acceleration in the resulting equation is ignored, the result is:

$$\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial}{\partial x}(\rho g H \frac{\partial \bar{\eta}}{\partial x}) + \frac{\partial}{\partial y}(\rho g H \frac{\partial \bar{\eta}}{\partial y}) = 0 \tag{7.45}$$

DRAFT

Diffraction

In a simplest case, we assume there are no currents. This means that $c_\sigma = 0$. Let denotes the propagation velocities in geographic and spectral spaces for the situation without diffraction as: $c_{x,0}$, $c_{y,0}$ and $c_{\theta,0}$. These are given by:

$$c_{x,0} = \frac{\partial \omega}{\partial k}\cos(\theta), \qquad c_{y,0} = \frac{\partial \omega}{\partial k}\sin(\theta), \qquad c_{\theta,0} = -\frac{1}{k}\frac{\partial \omega}{\partial h}\frac{\partial h}{\partial n} \tag{7.46}$$

where k is the wave number and n is perpendicular to the wave ray. We consider the following eikonal equation:

$$K^2 = k^2(1+\delta) \tag{7.47}$$

with δ denoting the diffraction parameter as given by:

$$\delta = \frac{\nabla(cc_g\nabla H_s)}{cc_g H_s} \tag{7.48}$$

Due to diffraction, the propagation velocities are given by:

$$c_x = c_{x,0}\bar{\delta}, \qquad c_y = c_{y,0}\bar{\delta}, \qquad c_\theta = c_{\theta,0}\bar{\delta} - \frac{\partial \bar{\delta}}{\partial x}c_{y,0} + \frac{\partial \bar{\delta}}{\partial y}c_{x,0} \tag{7.49}$$

where $\bar{\delta} = \sqrt{1+\delta}$.

7.5 Numerical implementation

The integration of the action balance equation has been implemented in SWAN with finite difference schemes in all five dimensions (time, geographic space and spectral space). In Delft3D-WAVE, SWAN is applied in a stationary mode so that time has been omitted from the equations. Below the propagation schemes in geographical and spectral space are briefly described.

The geographic space is discretised with a rectangular grid with constant resolutions Δx and Δy in x- and y-direction respectively (in fact, this rectangular grid is a special case of the curvi-linear grid that has been programmed in SWAN. The spectrum in the model is discretised with a constant directional resolution $\Delta\theta$ and a constant relative frequency resolution $\Delta\sigma/\sigma$ (logarithmic frequency distribution). For reasons of economy, an option is available to compute only wave components travelling in a pre-defined directional sector ($\theta_{min} < \theta < \theta_{max}$; e.g., those components that travel shorewards within a limited directional sector). The discrete frequencies are defined between a fixed low-frequency cut-off and a fixed high-frequency cut-off (the prognostic part of the spectrum). For these frequencies the spectral density is unconstrained. Below the low-frequency cut-off (typically $f_{min} = 0.04$ Hz for field conditions) the spectral densities are assumed to be zero. Above the high-frequency cut-off (typically 1 Hz for field conditions) a diagnostic f^{-m} tail is added (this tail is used to compute non-linear wave-wave interactions at the high frequencies and to compute integral wave parameters). The reason for using a fixed high-frequency cut-off rather than a dynamic cut-off frequency that depends on the wind speed or on the mean frequency, as in the WAM and WAVEWATCH III model, is that in coastal regions mixed sea states with rather different characteristic frequencies may occur. For instance, a local wind may generate a very young sea behind an island, totally unrelated to (but superimposed on) a simultaneously occurring swell. In such cases a dynamic cut-off frequency may be too low to properly account for the locally generated sea state. Based on physical arguments the value of m (the power in the above expression of the spectral tail) should be between 4 and 5 (Phillips, 1985). In SWAN $m = 4$ if the wind input formulation of Komen *et al.* (1984) is used (cf. WAM Cycle 3) and $m = 5$ if the wind input formulation of Janssen (1991a) is used (cf. WAM Cycle 4).

DRAFT

7.5.1 Propagation

The numerical schemes in SWAN have been chosen on the basis of robustness, accuracy and economy. Since the nature of the basic equation is such that the state in a grid point is determined by the state in the up-wave grid points, the most robust scheme would be an implicit upwind scheme (in both geographic and spectral space). The adjective "implicit" is used here to indicate that all derivatives of action density (x or y) are formulated at one computational level, i_x or i_y, except the derivative in the integration dimension for which also the previous or up-wave level is used (x or y in stationary mode). For such a scheme the values of space steps, Δx and Δy would be mutually independent. An implicit scheme would also be economical in the sense that such a scheme is unconditionally stable. It permits relatively large time steps in the computations (much larger than for explicit schemes in shallow water). Several years of experience in using the second-generation HISWA shallow water wave model (Holthuijsen *et al.*, 1989) has shown that for coastal regions a first-order upwind difference scheme in geographic space is usually accurate enough. This experience, together with test computations with SWAN has also shown that in spectral space a higher accuracy than that of a first-order upwind scheme is required. This can be achieved by supplementing such a scheme with a second-order central approximation (more economic than a second-order upwind scheme). For SWAN therefore, implicit upwind schemes in both geographic and spectral space have been chosen, supplemented with a central approximation in spectral space.

The fact that in geographic space, the state in a grid point is determined by the state in the up-wave grid points (as defined by the direction of propagation), permits a decomposition of the spectral space into four quadrants. In each of the quadrants the computations can be carried out independently from the other quadrants except for the interactions between them due to refraction and non-linear wave-wave interactions (formulated in corresponding boundary conditions between the quadrants). The wave components in SWAN are correspondingly propagated in geographic space with the first-order upwind scheme in a sequence of four forward-marching sweeps (one per quadrant). To properly account for the boundary conditions between the four quadrants, the computations are carried out iteratively at each time step. The discretization of the action balance equation is (for positive propagation speeds; including the computation of the source terms but ignoring their discretisation):

$$\begin{aligned}
&\left[\frac{[c_x N]_{i_x} - [c_x N]_{i_x-1}}{\Delta x}\right]^n_{i_y,i_\sigma,i_\theta} + \left[\frac{[c_y N]_{i_y} - [c_y N]_{i_y-1}}{\Delta y}\right]^n_{i_x,i_\sigma,i_\theta} \\
&+ \left[\frac{(1-\nu)[c_\sigma N]_{i_\sigma+1} + 2\nu[c_\sigma N]_{i_\sigma} - (1+\nu)[c_\sigma N]_{i_\sigma-1}}{2\Delta\sigma}\right]^n_{i_x,i_y,i_\theta} \\
&+ \left[\frac{(1-\eta)[c_\theta N]_{i_\theta+1} + 2\eta[c_\theta N]_{i_\theta} - (1+\eta)[c_\theta N]_{i_\theta-1}}{2\Delta\theta}\right]^n_{i_x,i_y,i_\sigma} = \left[\frac{S}{\sigma}\right]^{n^*}_{i_x,i_y,i_\sigma,i_\theta}
\end{aligned} \tag{7.50}$$

where i_x, i_y, i_σ and i_θ are grid counters and Δx, Δy, $\Delta\sigma$ and $\Delta\theta$ are the increments in geographic space and spectral space respectively. The iterative nature of the computation is indicated with the iteration index n (the iteration index for the source terms n^* is equal to n or $n-1$, depending on the source term, see below). Because of these iterations, the scheme is also approximately implicit for the source terms. For negative propagation speeds, appropriate + and - signs are required in Eq. 7.50.

The coefficients ν and η determine the degree to which the scheme in spectral space is upwind or central. They thus control the numerical diffusion in frequency and directional space,

respectively. A value of $\nu = 0$ or $\eta = 0$ corresponds to central schemes which have the largest accuracy (numerical diffusion $\gg 0$). Value of $\nu = 1$ or $\eta = 1$ correspond to upwind schemes which are somewhat more diffusive and therefore less accurate but more robust. If large gradients of the action density in frequency space or directional space are present, numerical oscillations can arise (especially with the central difference schemes) resulting in negative values of the action density. In each sweep such negative values are removed from the two-dimensional spectrum by setting these values equal to zero and re-scaling the remaining positive values such that the frequency-integrated action density per spectral direction is conserved. The depth derivatives and current derivatives in the expressions of c_σ and c_θ are calculated with a first-order upwind scheme. For very strong refraction the value of c_θ is reduced in each grid point and for each wave component individually with the square of the fraction of the grid spacing over which $kd < 3.0$.

The propagation scheme is implicit as the derivatives of action density (in x or y) at the computational level (i_x or i_y, respectively) are formulated at that level except in the integration dimension (x or y; depending on the direction of propagation) where also the up-wave level is used. The values of Δx and Δy are therefore still mutually independent.

The boundary conditions in SWAN, both in geographic space and spectral space are fully absorbing for wave energy that is leaving the computational domain or crossing a coast line. The incoming wave energy along open geographic boundaries needs to be prescribed by you. For coastal regions such incoming energy is usually provided only along the deep-water boundary and not along the lateral geographic boundaries (i.e., the spectral densities are assumed to be zero). This implies that such erroneous lateral boundary conditions are propagated into the computational area. The affected areas are typically triangular regions with the apex at the corners between the deep-water boundary and the lateral boundaries, spreading towards shore at an angle of 30° to 45° (for wind sea conditions) on either side of the deep-water mean wave direction (less for swell conditions; this angle is essentially equal to the one-sided width of the directional distribution of the incoming wave spectrum). For this reason the lateral boundaries should be sufficiently far away from the area of interest to avoid the propagation of this error into the area.

DRAFT

DRAFT

References

Abreu, M., A. Larraza and E. Thornton, 1992. "Nonlinear transformation of directional wave spectra in shallow water." *Journal of Geophysical Research* 97: 15579-15589.

Alves, J. and M. Banner, 2003. "Performance of a saturation-based dissipation-rate source term in modelling the fetch-limited evolution of wind waves." *J. Phys. Oceanogr.* 33: 1274-1298.

Arcilla, A. and C. Lemos, 1990. *Surf-Zone Hydrodynamics*. Centro Internacional de Métodos Numéricos en Ingenieria, Barcelona.

Arcilla, A., B. Roelvink, B. O'Connor, A. Reniers and J. Jimenez, 1994. "The Delta flume '93 experiment." In *Proceedings Coastal Dynamics Conference '94*, pages 488-502.

Battjes, J. and S. Beji, 1992. "Breaking waves propagating over a shoal." In *Proceedings 23rd International Conference Coastal Engineering, ASCE*, pages 42-50.

Battjes, J. and J. Janssen, 1978. "Energy loss and set-up due to breaking of random waves,." In *Proceedings 16th International Conference Coastal Engineering, ASCE*, pages 569-587.

Battjes, J. and M. Stive, 1985. "Calibration and verification of a dissipation model for random breaking waves." *Journal of Geophysical Research* 90 (C5): 9159-9167.

Beji, S. and J. Battjes, 1993. "Experimental investigation of wave propagation over a bar." *Coastal Engineering* 19: 151-162.

Bertotti, L. and L. Cavaleri, 1994. "Accuracy of wind and wave evaluation in coastal regions." In *Proceedings 24th International Conference Coastal Engineering, ASCE*, pages 57-67.

Booij, N., L. H. Holthuijsen and P. H. M. de Lange, 1992. "The penetration of short-crested waves through a gap." In *Proceedings 23rd International Conference Coastal Engineering, Venice 4-9 Oct., 1992, New York, 1993*, pages 1044-1052.

Booij, N., R. Ris and L. Holthuijsen, 1999. "A third-generation wave model for coastal regions, Part I, Model description and validation." *Journal of Geophysical Research* 104 (C4): 7649-7666.

Bouws, E. and G. Komen, 1983. "On the balance between growth and dissipation in an extreme, depth-limited wind-sea in the southern North Sea." *Journal of Physical Oceanography* 13: 1653-1658.

Cavaleri, L. and P. Malanotte-Rizzoli, 1981. "Wind wave prediction in shallow water: Theory and applications." *Journal of Geophysical Research* 86 (C11): 10 961-10 973.

Chen, Y. and R. Guza, 1997. "Modelling of breaking surface waves in shallow water." *Journal of Geophysical Research* 102 (C11): 25 035-25 046.

Collins, J., 1972. "Prediction of shallow water spectra." *Journal of Geophysical Research* 77 (15): 2693-2707.

Delft3D-FLOW, 2013. *Delft3D-FLOW User Manual*. Deltares, 3.14 ed.

Delft3D-IM, 2013. *Delft3D Installation Manual*. Deltares, 4.01 ed.

Dingemans, M. W., 1997. *Water Wave Propagation over Uneven Bottoms, Vol. 1 and 2*. Advanced Series on Ocean Engineering, Vol. 13. World Scientific, London.

Dingemans, M. W., A. C. Radder and H. J. de Vriend, 1987. "Computation of the driving forces of wave-induced currents." *Coastal Engineering* 11: 539-563.

DRAFT

Eldeberky, Y. and J. Battjes, 1995. "Parameterization of triad interactions in wave energy models, Gdansk, Poland." In *Proceedings Coastal Dynamics Conference '95*, pages 140-148.

Eldeberky, Y. and J. A. Battjes, 1996. "Spectral modelling of wave breaking: Application to Boussinesq equations." *Journal of Geophysical Research* 101 (C1): 1253-1264.

Elgar, S., R. Guza, B. Raubenheimer, T. Herbers and E. Gallagher, 1997. "Spectral evolution of shoaling and breaking waves on a barred beach." *Journal of Geophysical Research* 102 (C7): 15797-15805.

Galvin, C., 1972. "Waves on beaches and resulting sediment transport." In *Wave breaking in shallow water*, pages 413-455. Academic Press Inc.

Goda, Y., H. Takeda and Y. Moriya, 1967. *Laboratory investigation of wave transmission over breakwaters*. Tech. Rep. 13, Rep. port and Harbour Res. Institution. (from Seelig, 1979).

GPP, 2013. *Delft3D-GPP User Manual*. Deltares, 2.14 ed.

Günther, H., S. Hasselmann and P. A. E. M. Janssen, 1992. *The WAM model Cycle 4 (revised version)*. Tech. Rep. 4, Deutsch. Klim. Rechenzentrum, Hamburg, Germany.

Hasselmann, K., 1974. "On the spectral dissipation of ocean waves due to whitecapping." *Boundary-Layer Meteorology* 6 (1-2): 107-127.

Hasselmann, K., T. P. Barnett, E. Bouws, H. Carlson, D. E. Cartwright, K. Enke, J. Ewing, H. Gienapp, D. E. Hasselmann, P. Kruseman, A. Meerburg, P. Müller, D. J. Olbers, K. Richter, W. Sell and H. Walden, 1973. "Measurements of wind wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP)." *Deutsche Hydrographische Zeitschrift* 8 (12).

Hasselmann, K. and J. Collins, 1968. "Spectral dissipation of finite-depth gravity waves due to turbulent bottom friction." *Journal of Marine Research* 26: 1-12.

Hasselmann, S. and K. Hasselmann, 1981. "A symmetrical method of computing the nonlinear transfer in a gravity-wave spectrum." *Hamburger Geophysikalische Einzelschriften* 52 (8): 138p. Serie A.

Hasselmann, S., K. Hasselmann, J. Allender and T. Barnett, 1985. "Computations and parameterizations of the nonlinear energy transfer in a gravity wave spectrum. Part II: Parameterizations of the nonlinear transfer for application in wave models." *Journal of Physical Oceanography* 15 (11): 1378-1391.

Holthuijsen, L., N. Booij and T. Herbers, 1989. "A prediction model for stationary, short-crested waves in shallow water with ambient currents." *Coastal Engineering* 13: 23-54.

Holthuijsen, L., N. Booij and R. Ris, 1993. "A spectral wave model for the coastal zone." In *Proceedings of 2nd International Symposium on Ocean Wave Measurement and Analysis, New Orleans*, pages 630-641.

Ilic, S., 1994. *The role of offshore breakwaters in the coastal defence: comparison of two measurement systems*. Tech. rep., University of Plymouth School of Civil and Structural Engineering.

Janssen, P., 1991a. "Quasi-linear theory of wind-wave generation applied to wave forecasting." *Journal of Physical Oceanography* 21: 1631-1642.

Janssen, P. A. E. M., 1991b. *Consequences of the effect of surface gravity waves on the mean air flow*. Tech. rep., International Union of Theor. and Appl. Mech. (IUTAM), Sydney, Australia. 193-198.

DRAFT

Jonsson, I., 1966. “Wave boundary layers and friction factors.” In *Proceedings 10th International Conference Coastal Engineering, ASCE*, pages 127-148.

Jonsson, I. and N. Carlsen, 1976. “Experimental and theoretical investigations in an oscillatory turbulent boundary layer.” *Journal of Hydraulic Research* 14: 45-60.

Jonsson, I. G., 1980. “A new approach to rough turbulent boundary layers.” *Ocean Engineering* 7: 109-152.

Kaminsky, G. and N. Kraus, 1993. “Evaluation of depth-limited wave breaking criteria.” In *Proceedings of 2nd International Symposium on Ocean Wave Measurement and Analysis*, pages 180-193. New Orleans.

Kirby, J. T. and T.-M. Chen, 1989. “Surface waves on vertically sheared flows: approximate dispersion relations.” *Journal of Geophysical Research* 94: 1013-1027.

Komen, G., L. Cavaleri, M. Donelan, K. Hasselmann, S. Hasselmann and P. Janssen, 1994. *Dynamics and Modelling of Ocean Waves*. Camebridge University Press.

Komen, G., S. Hasselmann and K. Hasselmann, 1984. “On the existence of a fully developed wind-sea spectrum.” *Journal of Physical Oceanography* 14: 1271-1285.

Kuik, A., G. van Vledder and L. Holthuijsen, 1988. “A method for the routine analysis of pitch-and-roll buoy wave data.” *Journal of Physical Oceanography* 18: 1020-1034.

Luo, W. and J. Monbaliu, 1994. “Effects of the bottom friction formulation on the energy balance for gravity waves in shallow water.” *Journal of Geophysical Research* 99 (C9): 18501-18511.

Madsen, O., Y.-K. Poon and H. Graber, 1988. “Spectral wave attenuation by bottom friction: Theory.” In *Proceedings 21th International Conference Coastal Engineering, ASCE*, pages 492-504.

Madsen, P. and O. Sørensen, 1993. “Bound waves and triad interactions in shallow water.” *Ocean Engineering* 20 (4): 359-388.

Mase, H. and J. Kirby, 1992. “Hybrid frequency-domain KdV equation for random wave transformation.” In *Proceedings 23th International Conference Coastal Engineering, ASCE*, pages 474-487.

Mastenbroek, C., G. Burger and P. Janssen, 1993. “The dynamical coupling of a wave model in a storm surge model through the atmospheric boundary layer.” *Journal of Physical Oceanography* 23: 1856-1866.

Mei, C., 1983. *The applied dynamics of ocean surface waves*. Wiley, New York.

Miles, J., 1957. “On the generation of surface waves by shear flows.” *Journal of Fluid Mechanics* 3: 185-204.

Nelson, R., 1987. “Design wave heights on very mild slopes: An experimental study.” *Transactions of the Institution of Engineers, Australia, Civil Engineering* 29: 157-161.

Nelson, R. C., 1994. “Depth limited wave heights in very flat regions.” *Coastal Engineering* 23: 43-59.

Phillips, N. A., 1957. “A co-ordinate system having some special advantages for numerical forecasting.” *Journal of Meteorology* 14: 184-185.

Phillips, O., 1985. “Spectral and statistical properties of the equilibrium range in wind-generated gravity waves.” *Journal of Fluid Mechanics* 156: 505-531.

DRAFT

Putnam, J. and J. Johnson, 1949. “The dissipation of wave energy by bottom friction.” *Transactions - American Geophysical Union* 30: 67-74.

QUICKIN, 2013. *Delft3D-QUICKIN User Manual*. Deltares, 4.00 ed.

QUICKPLOT, 2013. *Delft3D-QUICKPLOT User Manual*. Deltares, 2.14 ed.

RGFGRID, 2013. *Delft3D-RGFGRID User Manual*. Deltares, 4.00 ed.

Ris, R., N. Booij and L. Holthuijsen, 1999. “A third-generation wave model for coastal regions, Part II: Verification.” *Journal of Geophysical Research* 104 (C4): 7649-7666.

Seelig, W., 1979. “Effects of breakwaters on waves: laboratory tests of wave transmission by overtopping.” In *Proceedings Conference Coastal Structures*, vol. 79:2, pages 941-961.

Shemdin, P., K. Hasselmann, S. Hsiao and K. Herterich, 1978. “Non linear and linear bottom interaction effects in shallow water.” In *Turbulent Fluxes through the Sea Surface Wave Dynamics and Prediction, NATO Conference Series*, no. 1 in V, pages 347-372.

Snyder, R., F. Dobson, J. Elliot and R. Long, 1981. “Array measurement of atmospheric pressure fluctuations above surface gravity waves.” *Journal of Fluid Mechanics* 102: 1-59.

SWAN, 2000. *SWAN Cycle III version 40.11 User Manual (not the short version)*. Delft University of Technology, Delft, The Netherlands, 0.00 ed.

Thornton, E. and R. Guza, 1983. “Transformation of wave height distribution.” *Journal of Geophysical Research* 88 (C10): 5925-5938.

Tolman, H., 1990. *Wind wave propagation in tidal seas*. Ph.D. thesis, Delft University of Technology, Department of Civil Engineering, The Netherlands.

Tolman, H. L., 1992a. “Effects of numerics on the physics in a third-generation windwave model.” *Journal of Physical Oceanography* 22: 1095-1111.

Tolman, H. L., 1992b. “An evaluation of expressions for the wave energy dissipation due to bottom friction in the presence of currents.” *Coastal Engineering* 16: 165-179.

Vincent, C., J. Smith and J. Davis, 1994. “Parameterization of wave breaking in models.” In M. Isaacson and M. Quick, eds., *Proceedings of International Symp.: Waves - Physical and Numerical Modelling*, vol. II, pages 753-762. University of British Columbia, Vancouver, Canada.

WAMDI group, 1988. “The WAM model a third generation ocean wave prediction model.” *Journal of Physical Oceanography* 18: 1775-1810.

Weber, S., 1989. *Surface gravity waves and turbulent bottom friction*. Ph.D. thesis, University of Utrecht, The Netherlands.

Weber, S. L., 1991a. “Bottom friction for wind sea and swell in extreme depth-limited situations.” *Journal of Physical Oceanography* 21: 149-172.

Weber, S. L., 1991b. “Eddy-viscosity and drag-law models for random ocean wave dissipation.” *Journal of Fluid Mechanics* 232: 73-98.

Westhuysen, A. J. Van der, 2007. *Advances in the spectral modelling of wind waves in the nearshore*. Ph.D. thesis, Delft University of Technology. Fac. of Civil Engineering.

Westhuysen, A. Van der, M. Zijlema and J. Battjes, 2007. *Nonlinear saturation-based whitecapping dissipation in SWAN for deep and shallow water*. Ph.D. thesis, Delft University of Technology.

DRAFT

Whitham, G., 1974. *Linear and nonlinear waves*. Wiley, New York.

Wilkens, 1999. *Bar Morphology Bornrif: modelling the evolution from 1982 to 1987*. Tech. rep., WL | Delft Hydraulics, Delft, The Netherlands. M.Sc.Thesis Univesity of Twente.

WL | Delft Hydraulics, 1999. *Modification first-guess SWAN and bench mark tests for SWAN*. Tech. Rep. H3515, WL | Delft Hydraulics, Delft, The Netherlands, Delft.

WL | Delft Hydraulics, 2000. *Physical formulations SWAN and data for validation*. Tech. Rep. H3528, WL | Delft Hydraulics, Delft, The Netherlands, Delft.

Wu, J., 1982. "Wind-stress coefficients over sea surface from breeze to hurricane." *Journal of Geophysical Research* 87 (C12): 9704-9706.

Young, I. R. and G. van Vledder, 1993. "A review of the central role of nonlinear interactions in wind-wave." *Philosophical transaction of the Royal Society London A* 342: 505-524.

DRAFT

DRAFT

A Files of Delft3D-WAVE

A.1 MDW-file

A.1.1 General description

File contents	The Master Definition WAVE file (MDW-file) is the input file for the wave simulation program.
Filetype	ASCII
File format	Free formatted
Filename	<*name*.mdw>
Generated	WAVE-GUI or manually offline

The Master Definition WAVE file (MDW-file) is the input file for the wave simulation program. It contains all the necessary data required for defining a model and running the simulation program. In the MDW-file you can define attribute files in which relevant data (for some parameters) are stored. This is especially useful when parameters contain a large number of data (e.g. time-dependent or space varying data). The user-definable attribute files are listed and described in Appendix A.

The MDW-file has the following general characteristics:

- ◇ Each line contains a maximum of 300 characters.
- ◇ Each set of input parameter(s) is preceded by a chapter name enclosed in square brackets (e.g. [WaveFileInformation]).
- ◇ Each input parameter is preceded by a `Keyword`.
- ◇ A `Keyword` is a combination of numerical and alpha-numerical characters, but starting with an alpha-numeric character, followed by an equal sign "=".

The MDW-file is an intermediate file between the WAVE-GUI and the WAVE simulation program. As it is an ASCII-file, it can be transported to an arbitrary hardware platform. Consequently, the WAVE simulation program and the WAVE-GUI do not necessarily have to reside on the same hardware platform.

Generally, you need not to bother about the internal layout or content of the MDW-file. It is, however, sometimes useful to be able to inspect the file and/or make small changes manually. Therefore the MDW-file is an ordinary ASCII-file which you can inspect and change with your favourite ASCII-editor.

The MDW-file is self contained, i.e. it contains all the necessary information about the model concerned. It can therefore be used as model archive by storing/printing the file.

Here we list all the possible chapters and keywords of the MDW-file:

Record description:

Keyword	Format	Description
WaveFileInformation		
`FileVersion`	string	should be 02.00
General		
`ProjectName`	C∗16	project name

continued on next page

∗ May be specified multiple times
+ Not supported by WAVE-GUI
R = Real; I = Integer; L = Logical; C = Character

continued from previous page

Keyword	Format	Description
ProjectNr	C∗4	project number
Description*	C∗72	description line
OnlyInputVerify	1 L	switch for input validation or simulation run: false = simulation run, or true = input validation only
SimMode	key-value	simulation mode: stationary, quasi-stationary, non-stationary
TimeStep	1 R	time step in case of non-stationary simulation
TScale+	1 R, optional	unit of time, default is 60.0)
FlowFile+	string	name of mdf-file containing FLOW input. If FlowFile is empty, FLOW is not running online. If FlowFile is non-empty, FLOW is running online.
FlowMudFile+	string	name of mdf-file containing FLOW input for the mud phase of a two phased FLOW model. If FlowMudFile is empty, MUD is not running online. If FlowMudFile is non-empty, MUD is running online.
FlowBedLevel	1 I	default usage of bed level from hydrodynamic computation by all domains: 0 = “don’t use”, 1 = “use but don’t extend”, 2 = “use and extend” if necessary. May be overruled by same keyword in group “domain”. Not relevant when FlowFile is empty; default: 0
FlowWaterLevel	1 I	See description of FlowBedLevel above.
FlowVelocity	1 I	See description of FlowBedLevel above.
FlowVelocityType	key-value	method of velocity computation (depth-averaged, surface-layer, wave-dependent; default: depth-averaged)
FlowWind	1 I	See description of FlowBedLevel above.
DirConvention	key-value	direction specification convention: nautical, cartesian
ReferenceDate	C∗10	reference date (string format: YYYY-MM-DD)
ObstacleFile	string	name of file containing obstacles
TSeriesFile	string	name of file containing time-dependent quantities
TimePntBlock	1 I, optional	number of table in TSeriesFile containing time points; only if TSeriesFile has been specified
MeteoFile*+	characters	Name of file containing meteo input
DirSpace	1 R, optional	default directional space: circle, sector
NDir	1 R, optional	default number of directional bins
StartDir	1 R, optional	default start direction in case of sector directional space
EndDir	1 R, optional	default end direction in case of sector directional space
NFreq	1 R, optional	default number of frequencies
FreqMin	1 R, optional	default minimum frequency
FreqMax	1 R, optional	default maximum frequency
WaterLevel	1 R	default water level
XVeloc	1 R	default velocity in x-direction
YVeloc	1 R	default velocity in y-direction
WindSpeed	1 R	default wind speed
WindDir	1 R	default wind direction
TimePoint* TimePoint should be specified if TimePntBlock is not included and not Online with FLOW.		
Time	1 R	time in minutes since refdate 0:00 hours
WaterLevel	1 R	water level at specified time point
XVeloc	1 R	velocity in x direction at specified time point
YVeloc	1 R	velocity in y direction at specified time point
WindSpeed	1 R	wind speed at specified time point
WindDir	1 R	wind direction at specified time point
Constants		
WaterLevelCorrection	1 R	Overall water level correction
Gravity	1 R	gravitational acceleration (default: 9.81 m/s^2)
WaterDensity	1 R	density of water (default: 1025 kg/m^3)
NorthDir	1 R	direction of north relative to x axis (default: 90°)
MinimumDepth	1 R	minimum water depth below which points are excluded from the computation (default: 0.05 m)
Processes		

continued on next page

* May be specified multiple times
+ Not supported by WAVE-GUI
R = Real; I = Integer; L = Logical; C = Character

DRAFT

continued from previous page

Keyword	Format	Description
GenModePhys	1 I	generation mode of physics: 1 for first-generation, 2 for second-generation, 3 for third-generation
WaveSetup	1 L	include wave setup (default: false)
Breaking	1 L	include wave breaking (default: true)
BreakAlpha	1 R	alpha coefficient for wave breaking (default: 1.0)
BreakGamma	1 R	gamma coefficient for wave breaking (default: 0.73)
Triads	1 L	include triads (default: false)
TriadsAlpha	1 R	alpha coefficient for triads (default: 0.1)
TriadsBeta	1 R	beta coefficient for triads (default: 2.2)
BedFriction	string	bed friction type (none, jonswap, collins, madsen et al., default: jonswap)
BedFricCoef	1 R	bed friction coefficient (default: 0.067 for jonswap, 0.015 for collins, 0.05 for madsen et al.)
Diffraction	1 L	include diffraction (default: true)
DiffracCoef	1 R	diffraction coefficient (default: 0.2)
DiffracSteps	1 I	number of diffraction smoothing steps (default: 5)
DiffracProp	1 L	include adaption of propagation velocities due to diffraction (default: true)
WindGrowth	1 L	include wind growth (default: true)
WhiteCapping	key-value	white capping: (Off, Komen, Westhuysen, default: Komen)
Quadruplets	1 L	include quadruplets (default: false)
Refraction	1 L	include refraction (default: true)
FreqShift	1 L	include frequency shifting in frequency space (default: true)
WaveForces	key-value	method of wave force computation (dissipation 3d, dissipation, radiation stresses $<$2013; default: dissipation 3d)
Numerics		
DirSpaceCDD	1 R	discretisation in directional space: 0 for central, 1 for upwind (default: 0.5)
FreqSpaceCSS	1 R	discretisation in frequency space: 0 for central, 1 for upwind (default: 0.5)
RChHsTm01	1 R	relative change of wave height or mean wave period with respect to local value (default: 0.02)
RChMeanHs	1 R	relative change of wave height with respect to model-wide average wave height (default: 0.02)
RChMeanTm01	1 R	relative change of mean wave period with respect to model-wide average mean wave period (default: 0.02)
PercWet	1 R	percentage of points included in simulation at which convergence criteria must be satisfied (default: 98%)
MaxIter	1 I	maximum number of iterations for convergence (default: 15)
Output		
TestOutputLevel	1 I	test output level (default: 0)
TraceCalls	1 L	trace subroutine calls (default: false)
UseHotFile	1 L	write and read hotstart files (default: false)
MapWriteInterval	1 R	interval for writing data to map file(s) in minutes
WriteCOM	1 L	write results to communication file(s) (default: false)
COMWriteInterval	1 R	interval for writing data to communication file(s) in minutes
AppendCOM	1 L	upon writing to communication file(s) overwrite the previous data (false) or append to the data series (true) (default: false)
MassFluxToCOM$^+$	1 L, optional	write mass fluxes due to wave to communication file(s) (default: true)
LocationFile	string, optional	file name of output locations
CurveFile	string, optional	file name of output curves
WriteTable	1 L	write tables for output locations (default: false)
WriteSpec1D	1 L	write 1D spectra for output locations (default: false)
WriteSpec2D	1 L	write 2D spectra for output locations (default: false)
Domain*		
Grid	string	file name of computational grid
BedLevelGrid	string	file name of bed level grid (default: equal to computational grid)
BedLevel	string	file name of bed level data
DirSpace	1 R	directional space: circle, sector

continued on next page

* May be specified multiple times
$^+$ Not supported by WAVE-GUI
R = Real; I = Integer; L = Logical; C = Character

continued from previous page

Keyword	Format	Description
NDir	1 R	number of directional bins
StartDir	1 R	start direction in case of sector directional space
EndDir	1 R	end direction in case of sector directional space
NFreq	1 R	number of frequencies
FreqMin	1 R	minimum frequency
FreqMax	1 R	maximum frequency
NestedInDomain	1 R	number of domain in which current domain is nested (required for domains 2 and following)
FlowBedLevel		See description of FlowBedLevel in group [General]
FlowWaterLevel		See description of FlowBedLevel in group [General]
FlowVelocity		See description of FlowBedLevel in group [General]
FlowVelocityType		See description of FlowBedLevel in group [General]
FlowWind		See description of FlowBedLevel in group [General]
MeteoFile*		Name of file containing meteo input
Output	1 L	write map file for current domain (default: true)
Boundary*		
Name	string	boundary name
Definition	key-value	definition type (orientation, grid-coordinates, xy-coordinates)
Orientation	key-value	boundary orientation in case of boundary definition by means of orientation (north, northwest, west, southwest, south, southeast, east, northeast)
DistanceDir	key-value	direction of distance measurements for boundary segments in case of boundary definition by means of orientation (clockwise, counter-clockwise; default: counter-clockwise)
StartCoordM	1 I	start m-coordinate of boundary in case of boundary definition by means of grid-coordinates
EndCoordM	1 I	end m-coordinate of boundary in case of boundary definition by means of grid-coordinates
StartCoordN	1 I	start n-coordinate of boundary in case of boundary definition by means of grid-coordinates
EndCoordN	1 I	end n-coordinate of boundary in case of boundary definition by means of grid-coordinates
StartCoordX	1 R	start x-coordinate of boundary in case of boundary definition by means of xy-coordinates
EndCoordX	1 R	end x-coordinate of boundary in case of boundary definition by means of xy-coordinates
StartCoordY	1 R	start y-coordinate of boundary in case of boundary definition by means of xy-coordinates
EndCoordY	1 R	end y-coordinate of boundary in case of boundary definition by means of xy-coordinates
SpectrumSpec	key-value	spectrum specification type (from file, parametric)
SpShapeType	key-value	spectrum shape type in case of parametric spectrum specification (jonswap, pierson-moskowitz, gauss)
PeriodType	key-value	wave period type in case of parametric spectrum specification (peak, mean)
DirSpreadType	key-value	directional spreading type in case of parametric spectrum specification (power, degrees)
PeakEnhancFac	1 R	peak enhancement factor in case of jonswap spectrum
GaussSpread	1 R	width of spectral distribution in case of gaussian spectrum
CondSpecAtDist*	1 R	distance along boundary at which boundary condition is specified, uniform boundary condition if not specified
WaveHeight*	1 R	wave height at specified distance or uniform value in case of parametric spectrum specification
Period*	1 R	wave period at specified distance or uniform valuein case of parametric spectrum specification
Direction*	1 R	wave direction at specified distance or uniform value in case of parametric spectrum specification
DirSpreading*	1 R	directional spreading at specified distance or uniform value in case of parametric spectrum specification
Spectrum*	string	file name containing spectrum (string) in case of spectrum specification from file

* May be specified multiple times
+ Not supported by WAVE-GUI
R = Real; I = Integer; L = Logical; C = Character

DRAFT

A.1.2 Offline calculation

When running WAVE offline using FLOW output, the following items are not supported by the WAVE-GUI and must be checked in the mdw-file with a text editor:

- ◇ The keyword `FlowFile` must be removed from the group `[General]`.
- ◇ A time point must be specified for each time for which a calculation must be performed

Example:

```
[Timepoint]
Time = 1440
[Timepoint]
Time = 1680
```

The specified time points must correspond with times written on the com-file.

A.2 Attribute files of Delft3D-WAVE

A.2.1 Introduction

In the following sections we describe the attribute files used in the input MDW-file of Delft3D-WAVE. Most of these files contain the quantities that describe one specific item, such as the bathymetry or the grid.

Most of the attribute files can be generated by the WAVE-GUI after defining an input scenario. Some files can only be generated by utility programs such as the curvilinear grid generated by RGFGRID . Still, we describe both types of files as it might be useful to know how the input data is structured to be able to generate (large) files.

For each file we give the following information (if relevant):

File content.
File type (free formatted, fix formatted or unformatted).
Filename and extension.
Generated by (i.e. how to generate the file).
Restrictions on the file content.
Example(s).

Remarks:

- ◇ The access mode of all attribute files is sequential.
- ◇ In the examples the file contents is printed in font Courier New 10 and comment (not included in the file) in font Times New Roman 9, unless stated explicitly differently.

DRAFT

A.2.2 Orthogonal curvilinear grid

File contents	The co-ordinates of the orthogonal curvilinear grid at the depth points.
Filetype	ASCII
File format	Free formatted
Filename	<*name*.grd>
Generated	RGFGRID

Record description:

Record	Record description
	Preceding description records, starting with an asterisk ($*$), will be ignored.
1	Record with `Co-ordinate System = Cartesian` or value `Spherical`
2	The number of grid *points* in m- and n-direction (2 integers).
3	Three real values (not used).
4 to K+3	A label and record number, the x-component of the world co-ordinates of all points in m-direction, starting with row 1 to row $nmax$, with as many continuation records as required by $mmax$ and the number of co-ordinates per record. The label and record number are suppressed on the continuation lines. This set of records is repeated for each row until $n = nmax$.
K+4 to 2K+3	A similar set of records for the y-component of the world co-ordinates.

K is the number of records to specify for all grid points a set of x- or y-co-ordinates.

Restrictions:
- ◇ The grid must be orthogonal.
- ◇ Input items in a record are separated by one or more blanks.

Example:

```
*
* Deltares, Delft3D-RGFGRID  Version 4.16.01.4531, Sep 30 2008, 23:32:27
* File creation date: 2008-10-01, 23:19:22
*
Coordinate System = Cartesian
       9       7
 0 0 0
 Eta=    1   0.0000000000000000E+00   1.0000000000000000E+02   2.000000...
             5.0000000000000000E+02   6.0000000000000000E+02   7.000000...
 Eta=    2   0.0000000000000000E+00   1.0000000000000000E+02   2.000000...
             5.0000000000000000E+02   6.0000000000000000E+02   7.000000...
 Eta=    3   0.0000000000000000E+00   1.0000000000000000E+02   2.000000...
             5.0000000000000000E+02   6.0000000000000000E+02   7.000000...
 Eta=    4   0.0000000000000000E+00   1.0000000000000000E+02   2.000000...
             5.0000000000000000E+02   6.0000000000000000E+02   7.000000...
 Eta=    5   0.0000000000000000E+00   1.0000000000000000E+02   2.000000...
             5.0000000000000000E+02   6.0000000000000000E+02   7.000000...
```

DRAFT

```
Eta=    6   0.00000000000000000E+00   1.00000000000000000E+02   2.000000...
            5.00000000000000000E+02   6.00000000000000000E+02   7.000000...
Eta=    7   0.00000000000000000E+00   1.00000000000000000E+02   2.000000...
            5.00000000000000000E+02   6.00000000000000000E+02   7.000000...
Eta=    1   1.00000000000000000E+02   1.00000000000000000E+02   1.000000...
            1.00000000000000000E+02   1.00000000000000000E+02   1.000000...
Eta=    2   2.00000000000000000E+02   2.00000000000000000E+02   2.000000...
            2.00000000000000000E+02   2.00000000000000000E+02   2.000000...
Eta=    3   3.00000000000000000E+02   3.00000000000000000E+02   3.000000...
            3.00000000000000000E+02   3.00000000000000000E+02   3.000000...
Eta=    4   4.00000000000000000E+02   4.00000000000000000E+02   4.000000...
            4.00000000000000000E+02   4.00000000000000000E+02   4.000000...
Eta=    5   5.00000000000000000E+02   5.00000000000000000E+02   5.000000...
            5.00000000000000000E+02   5.00000000000000000E+02   5.000000...
Eta=    6   6.00000000000000000E+02   6.00000000000000000E+02   6.000000...
            6.00000000000000000E+02   6.00000000000000000E+02   6.000000...
Eta=    7   7.00000000000000000E+02   7.00000000000000000E+02   7.000000...
            7.00000000000000000E+02   7.00000000000000000E+02   7.000000...
```

A.2.3 Time-series for wave boundary conditions

File contents	Time-series for wave boundary conditions.
Filetype	ASCII
File format	Fix format for header information; free format for time-series data.
Filename	<*name*.bcw>
Generated	FLOW-GUI, program Delft3D-NESTHD or manually offline

Record description:

Keyword	**Description**
`location`	location name (quoted string)
`time-function`	time function type (quoted string: "non-equidistant")
`reference-time`	reference time (yyyymmdd integer or quoted string: "from model")
`time-unit`	time unit (quoted string: "decades", "years", "days", "hours", "minutes", "seconds", "ddhhmmss", "absolute")
`interpolation`	interpolation type (quoted string: "linear" or "block")
`parameter & unit`	parameter name & unit

A.2.4 Obstacle file

File contents	Name of the polyline with obstacles.
Filetype	ASCII
File format	Fix formatted for text variables, free formatted for real and integer values.
Filename	<*name*.obs>
Generated	QUICKIN as land boundary, or manually offline

Record description:

A header block containing information about versions, and the name of the polyline file.

For each observation area the details.

DRAFT

Keyword	Format	Description
ObstacleFileInformation		
FileVersion	string	version number of <∗.obs> file
PolylineFile	string	name of polyline file with polylines defining obstacles
Obstacle*		
Name	string	name of obstacle in polyline file
Type	key-value	type of obstacle (sheet, dam)
TransmCoef	1 real	transmission coefficient in case of sheet obstacle
Height	1 real	dam height in case of dam obstacle
Alpha	1 real	alpha in case of dam obstacle
Beta	1 real	beta in case of dam obstacle
Reflections	key-value	type of reflections (no, specular, diffuse)
ReflecCoef	1 real	reflection coefficient if reflections are activated

* May be specified multiple times

Restriction:

- The maximum record length in the file is 132.

Example:

The number of obstacles is 2. They are called ‘Breakwater West’, ’Breakwater East 2’ and ’Breakwater East 1’

```
[ObstacleFileInformation]
   FileVersion  = 02.00
   PolylineFile = breakwater.pol
[Obstacle]
   Name         = Breakwater West
   Type         = dam
   Height       =  0.0000000e+000
   Alpha        =  2.5999999e+000
   Beta         =  1.5000001e-001
   Reflections  = no
[Obstacle]
   Name         = Breakwater East 1
   Type         = dam
   Height       =  0.0000000e+000
   Alpha        =  2.5999999e+000
   Beta         =  1.5000001e-001
   Reflections  = no
[Obstacle]
   Name         = Breakwater East 2
   Type         = dam
   Height       =  0.0000000e+000
   Alpha        =  2.5999999e+000
   Beta         =  1.5000001e-001
   Reflections  = no
```

Example polyline file:

```
Breakwater West
```

DRAFT

```
    7      2
  1.9174138E+05   6.0961231E+05
  1.9190197E+05   6.1048831E+05
  1.9242755E+05   6.1140806E+05
  1.9321591E+05   6.1228400E+05
  1.9422327E+05   6.1301400E+05
  1.9536202E+05   6.1358338E+05
  1.9655916E+05   6.1394831E+05
Breakwater East 1
     2      2
  2.0846027E+05   6.0775812E+05
  2.0838540E+05   6.0968968E+05
Breakwater East 2
     2      2
  2.1022712E+05   6.0998915E+05
  2.1031696E+05   6.0765331E+05
```

A.2.5 Segment file

File contents	The coordinates of one or more polylines. Each polyline (piecewise linear) is written in a single block of data.
Filetype	ASCII
File format	Free formatted
Filename	<*name*.pol>
Generated	RGFGRID, QUICKIN, Delta Shell, etc

Record description:

Record	Record description
	Preceding description records, starting with an asterisk ($*$), and will be ignored.
1	A non blank character string, starting in column one.
2	Two integers N_r, N_c representing the numbers of rows and number of columns for this block of data.
	Two reals representing the x, y or λ, ϕ-coordinate, followed by remaining data values at that location (if $N_c > 2$).

Example:

```
*
* Polyline L007
*
L007
6  2
            132400.0    549045.0
            132345.0    549030.0
            132165.0    549285.0
            131940.0    549550.0
            131820.0    549670.0
            131585.0    549520.0
*
* Polyline L008
*
L008
4  2
```

DRAFT

```
            131595.0    549685.0
            131750.0    549865.0
            131595.0    550025.0
            131415.0    550175.0
*
* Polyline L009
*
L009
6  2
            131595.0    549655.0
            148975.0    564595.0
            150000.0    564935.0
            152105.0    565500.0
            153150.0    566375.0
            154565.0    567735.0
```

A.2.6 Depth file

File contents	The bathymetry in the model area, represented by depth values (in metres) for all grid points.
Filetype	ASCII
File format	Free formatted or unformatted
Filename	<*name*.dep>
Generated	FLOW-GUI (only for uniform depth values). Offline with QUICKIN and data from digitised charts or GIS-database.

Record description:

Filetype	Record description
Free formatted	Depth values per row, starting at N = 1 to N = Nmax, separated by one or more blanks. The number of continuation lines is determined by the number of grid points per row (Mmax) and the maximum record size of 132.
Unformatted	Mmax depth values per row for N = 1 to N = Nmax.

Restrictions:

- ◇ The file contains one M and N line more than the grid dimension.
- ◇ The maximum record length in the free formatted file is 132.
- ◇ Depth values from the file will not be checked against their domain.
- ◇ The input items are separated by one or more blanks (free formatted file only).
- ◇ The default missing value is: -999.0

Example:

File containing $16 * 8$ data values for a model area with $15 * 7$ grid points (free formatted file).

```
  1.0    2.0    3.0    4.0   -5.0   -5.0   -5.0    8.0    9.0   10.0   11.0
 12.0   13.0   14.0   -5.0 -999.0
  3.0    4.0    5.0    6.0    7.0   -6.0   -6.0   10.0   11.0   12.0   13.0
 14.0   15.0   16.0   17.0 -999.0
  5.0    6.0    7.0    8.0    9.0   10.0   -7.0   12.0   13.0   14.0   15.0
 16.0   17.0   18.0   19.0 -999.0
  7.0    8.0    9.0   10.0   11.0   12.0   13.0   14.0   15.0   16.0   17.0
 18.0   19.0   -7.0   19.0 -999.0
  9.0   10.0   11.0   12.0   13.0   14.0   15.0   16.0   17.0   18.0   19.0
```

DRAFT

```
  20.0   19.0   18.0   17.0 -999.0
  -7.0   12.0   13.0   14.0   15.0   16.0   17.0   18.0   19.0   20.0   19.0
  18.0   17.0   16.0   15.0 -999.0
  -8.0   -8.0   15.0   16.0   17.0   18.0   19.0   20.0   19.0   18.0   17.0
  16.0   15.0   14.0   13.0 -999.0
-999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0
-999.0 -999.0 -999.0 -999.0 -999.0
```

The resulting 2D-matrix for the depth values is then (for simplicity all values are here transformed into integers, in reality this does not occur):

```
N-direction
↑   8 | -9  -9  -9  18  19  20  19  18  17  16  15  14  13  12  -9  -9
    7 | -8  -8  15  16  17  18  19  20  19  18  17  16  15  14  13  -8
    6 | -7  12  13  14  15  16  17  18  19  20  19  18  17  16  15  14
    5 |  9  10  11  12  13  14  15  16  17  18  19  20  19  18  17  16
    4 |  7   8   9  10  11  12  13  14  15  16  17  18  19  -7  19  18
    3 |  5   6   7   8   9  10  -7  12  13  14  15  16  17  18  19  20
    2 |  3   4   5   6   7  -6  -6  10  11  12  13  14  15  16  17  -6
    1 |  1   2   3   4  -5  -5  -5   8   9  10  11  12  13  14  -5  -5
      +-------------------------------------------------------------------
      |  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
                                                        → M-direction
```

A.2.7 Space-varying bottom friction (not yet implemented for Delft3D-WAVE)

File contents: Bottom friction coefficients values (induced by waves) for all grid points, starting from row number (y-direction) 1 for all points in the x-direction (1 to MMAX), until the last row number (NMAX). Note that for the bottom friction values also a constant value over the entire computational area can be applied (see section 4.5.7.3).

File type: free formatted / unformatted.

Restrictions: maximum record length in the (free) formatted file is 132. Bottom friction coefficients values from the file will not be checked against the ranges specified in section 4.5.7.3 (domain of input parameters).

Example: (formatted file)

```
0.01   0.01   0.02   0.03
0.01   0.01   0.02   0.03
0.012  0.012  0.011  0.03
0.013  0.013  0.013  0.03
0.014  0.014  0.013  0.03
```

The resulting 2D-matrix for the bottom friction coefficients values:

```
N-direction
↑   8 |
    7 |
    6 | 0.014  0.014  0.013  0.03
    5 | 0.013  0.013  0.013  0.03
    4 | 0.012  0.012  0.011  0.03
    3 | 0.011  0.011  0.01   0.03
    2 | 0.01   0.01   0.02   0.03
    1 | 0.01   0.01   0.02   0.03
      +-------------------------------------------------------------------
      |  1     2      3      4    5  6  7  8  9  10  11  12  13  14  15  16
                                                        → M-direction
```

DRAFT

A.2.8 Wave boundary conditions

In Delft3D-WAVE the users could choose different sets of wave boundary conditions and wind conditions. However not all the features could be specified by the GUI. The functionalities could be used by adding keywords in <mdw>-file.

In the following subsections, 4 options are described:

1 Time-varying and uniform wave conditions in <wavecon.*rid*>.
2 Time-varying and space-varying wave boundary conditions using <bcw>-files
3 Space-varying wave boundary conditions using for UNIBEST coupling (<md-vwac> file)
4 Space-varying wave boundary conditions: Spectral input and output files

A.2.8.1 Time-varying and uniform wave conditions in <wavecon.*rid*> file

In some cases where e.g. the morphology is event-driven or design conditions for a structure are needed, a set of different wave conditions are to be calculated. These wave conditions can be specified in an additional file, called <wavecon.*rid*> (rid=runid of the <mdw>-file). This file can only be used when constant parametric boundary conditions are prescribed in the wave model. If other boundary conditions are specified, these will be adjusted into constant parametric boundary conditions. To use this *Wavecon* option, just simply add the <wavecon.*rid*> file to the working directory and the system will use the file automatically.

A WAVE computation is always performed on a certain time point (based on the reference date). If a <wavecon.*rid*> file exists in the working directory, it will get its wave boundary conditions (including wind and water level) from that file. The boundary condition values in the default <*rid*.mdw> file will not be used then. When the time point of the wave computation lies between two prescribed time points in the <wavecon.*rid*> file, it will interpolate the wave, wind and water level conditions between these two time points.

Remarks:

- ◇ If the wind speed is prescribed as 0 m/s, wind will not be taken into account in the wave computation.
- ◇ If the time point of the wave computation lies before the first prescribed time field in the <wavecon.*rid*> file, it will use the conditions of this first field.
- ◇ If a mean period is chosen in the default <*rid*.mdw> file, this period will be modified into the peak period (the value of the period will remain the same).
- ◇ If a variable boundary condition is chosen in the default <*rid*.mdw> file, this condition will be modified into a constant condition along the whole boundary.
- ◇ The defined wave boundary conditions are overruled by the prescribed wave conditions in the <wavecon.∗> file.

File contents:	List of wave and wind conditions
File type:	free formatted/unformatted.
Restrictions:	maximum record length in the (free) formatted file is 132.
Example:	formatted file of a <wavecon.*rid*>

```
* Itdate   Hs     Tp     Dir(°)   ms   wl     windspeed   wind dir.(°)
BL01
3 8 * number of rows    number of columns
0          0.01   1.0    270      10   0      0.0         270
60         1.00   7.0    270      4    1.26   10.0        270
240        0.01   10.0   270      10   0.70   5.0         270
```

Description of parameters:

Itdate [min]	Time point after reference date in minutes; should be given in minutes after the reference date (ITDATE), specified in the <*rid*.mdw> file.
H_s [m]	Significant wave height in metres; this value will be prescribed on all specified wave boundaries.
T_p [s]	Peak period of the energy spectrum. This value will be prescribed on all specified wave boundaries.
Dir [°]	Mean wave direction according to the Nautical or Cartesian convention (in degrees). This value will be prescribed on all specified wave boundaries.
ms [-] or [°]	Width energy distribution. This is the directional standard deviation in power or in degrees. If the option *Degrees* is chosen in the sub-window *Spectral space*, it is in degree. If the option *Cosine power* is chosen in the same above sub-window, it is in the power m.
Water level [m]	The additional water level over the entire wave model. The water level is measured positively upward from the same datum from which the bottom levels are taken.
Wind speed [m/s]	Wind velocity at 10 m elevation.
Wind direction [°]	Wind direction at 10 m elevation according to the convention, specified in the sub-window **Constants**.

Remarks:

- ◇ The defined wave boundary conditions in the mdw file are overruled by the prescribed wave conditions in the <wavecon.∗> file.
- ◇ If wavecon or <md-vwac> file is used as wave boundary condition, the width energy distribution *ms* is set (overwritten) to be power.

A.2.8.2 Time-varying and space-varying wave boundary conditions using BCW files

In Delft3D-WAVE, time series of wave boundary conditions have been implemented which are not able to be set in GUI yet. The users can include the keywords *TSeriesFile* in Datagroup *General* in MDW-file. The format of BCW-file refer to the section A.2.3. The segments of boundary conditions could be set using the keywords `CondSpecAtDist` in Datagroup *Boundary* in MDW-file. If the wave computations are carried out at multiple time points, the time point could be specified in Datagroup *Timepoint* in MDW-file.

The following examples showed different scenarios of spatial-varying and time-varying wave boundnary conditions. It is a stand-alone wave model with 2 boundaries, i.e., Boundary West and Boundary South. The Boundary West is devided into 6 segments and the Boundary South is devided into 9 segments. For each segments, different parameters such as Wave Height, Period, Direction, Dirspreading could be defined at different time point in the BCW-file.

The 3 examples show the following 3 scenarios:

1 Multiple time points and spatial uniform wave boundary conditions.
2 One/multiple time points and space-varying wave boundary conditions
3 Multiple time points and space-varying wave boundary conditions, with time-varying but spatial uniform wind field

DRAFT

Example 1

If one would like to have a wave model with uniform wave boundary conditions along one boundary line for multiple time points, one should add them to Datagroup *General* as follows:

```
[WaveFileInformation]
   FileVersion          = 02.00
[General]
   ProjectName          = Carrara
   ProjectNr            = 001
   Description          =
   Description          = Carrara test run
   OnlyInputVerify      = false
   SimMode              = stationary
   DirConvention        = nautical
   ReferenceDate        = 2006-01-05
   TSeriesFile          = timeseries.bcw
   WindSpeed            =  2.0
   WindDir              =  2.0
...
```

In Datagroup *TimePoint* the following should be added:

```
...
[TimePoint]
   Time                 =  6.0000000e+001
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
[TimePoint]
   Time                 =  1.2000000e+002
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
[TimePoint]
   Time                 =  1.8000000e+002
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
[TimePoint]
   Time                 =  2.4000000e+002
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
...
```

In Datagroup *Boundary* the following should be added:

```
...
[Boundary]
   Name                 = Boundary West
   Definition           = xy-coordinates
   StartCoordX          =  5.0000000e+005
   EndCoordX            =  5.0000000e+005
   StartCoordY          =  4.9274090e+006
   EndCoordY            =  4.7885805e+006
   SpectrumSpec         = parametric
   SpShapeType          = jonswap
   PeriodType           = peak
   DirSpreadType        = power
```

DRAFT

```
   PeakEnhanceFac       =  3.3000000e+000
   GaussSpread          =  9.9999998e-003
[Boundary]
   Name                 = Boundary South
   Definition           = xy-coordinates
   StartCoordX          =  5.0000000e+005
   EndCoordX            =  6.2226400e+005
   StartCoordY          =  4.7608150e+006
   EndCoordY            =  4.7608150e+006
   SpectrumSpec         = parametric
   SpShapeType          = jonswap
   PeriodType           = peak
   DirSpreadType        = power
   PeakEnhanceFac       =  3.3000000e+000
   GaussSpread          =  9.9999998e-003
...
```

The <bcw>-file, which is defined in section A.2.3, for the uniform boundaries with multiple time points should be then:

```
...
location             'Boundary West       '
time-function        'non-equidistant'
reference-time       20060105
time-unit            'minutes'
interpolation        'linear'
parameter            'time                ' unit '[min]'
parameter            'WaveHeight'           unit '[m]'
parameter            'Period'               unit '[s]'
parameter            'Direction'            unit '[N^o]'
parameter            'DirSpreading'         unit '[-]'
   0.00  5.5300 8.2400 -171.0700 2.0000
  60.00  3.5300 8.2400 -171.0700 2.0000
 120.00  1.5300 8.2400 -171.0700 2.0000
 180.00  3.5300 8.2400 -171.0700 2.0000
 240.00  1.5300 8.2400 -171.0700 2.0000
location             'Boundary South      '
time-function        'non-equidistant'
reference-time       20060105
time-unit            'minutes'
interpolation        'linear'
parameter            'time                ' unit '[min]'
parameter            'WaveHeight'           unit '[m]'
parameter            'Period'               unit '[s]'
parameter            'Direction'            unit '[N^o]'
parameter            'DirSpreading'         unit '[-]'
   0.00  1.2700 8.4700 -147.8800 2.0000
  60.00  3.2700 8.4700 -147.8800 2.0000
 120.00  1.2700 8.4700 -147.8800 2.0000
 180.00  3.2700 8.4700 -147.8800 2.0000
 240.00  3.2700 8.4700 -147.8800 2.0000
```

Example 2

If one would like to have a wave model with space-varying wave boundary conditions, one should add them to Datagroup *General* as follows:

```
[WaveFileInformation]
   FileVersion          = 02.00
[General]
   ProjectName          = Carrara
   ProjectNr            = 001
   Description          =
```

DRAFT

```
   Description          = Carrara test run
   OnlyInputVerify      = false
   SimMode              = stationary
   DirConvention        = nautical
   ReferenceDate        = 2006-01-05
   TSeriesFile          = timeseries.bcw
   WindSpeed            =  2.0
   WindDir              =  2.0
...
```

In Datagroup *TimePoint* the following should be added:

```
...
[TimePoint]
   Time                 =  6.0000000e+001
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
...
```

In Datagroup *Boundary* the following should be added:

```
...
[Boundary]
   Name                 = Boundary West
   Definition           = xy-coordinates
   StartCoordX          =  5.0000000e+005
   EndCoordX            =  5.0000000e+005
   StartCoordY          =  4.9274090e+006
   EndCoordY            =  4.7885805e+006
   SpectrumSpec         = parametric
   SpShapeType          = jonswap
   PeriodType           = peak
   DirSpreadType        = power
   PeakEnhanceFac       =  3.3000000e+000
   GaussSpread          =  9.9999998e-003
   CondSpecAtDist       =  2.7765670e+004
   CondSpecAtDist       =  5.5531340e+004
   CondSpecAtDist       =  6.3297008e+004
   CondSpecAtDist       =  8.3297008e+004
   CondSpecAtDist       =  1.1106268e+005
   CondSpecAtDist       =  1.3882834e+005
[Boundary]
   Name                 = Boundary South
   Definition           = xy-coordinates
   StartCoordX          =  5.0000000e+005
   EndCoordX            =  6.2226400e+005
   StartCoordY          =  4.7608150e+006
   EndCoordY            =  4.7608150e+006
   SpectrumSpec         = parametric
   SpShapeType          = jonswap
   PeriodType           = peak
   DirSpreadType        = power
   PeakEnhanceFac       =  3.3000000e+000
   GaussSpread          =  9.9999998e-003
   CondSpecAtDist       =  0.0000000e+000
   CondSpecAtDist       =  1.0000000e+003
   CondSpecAtDist       =  1.0000000e+004
   CondSpecAtDist       =  2.0377330e+004
   CondSpecAtDist       =  4.0754660e+004
   CondSpecAtDist       =  6.1131988e+004
   CondSpecAtDist       =  8.1509320e+004
```

DRAFT

```
   CondSpecAtDist       =  1.0188665e+005
   CondSpecAtDist       =  1.2226398e+005
...
```

The <bcw>-file, which is defined in section A.2.3, should be like:

```
...
location             'Boundary West      '
time-function        'non-equidistant'
reference-time       20060105
time-unit            'minutes'
interpolation        'linear'
parameter            'time               '                  unit '[min]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'Period'                               unit '[s]'
parameter            'Period'                               unit '[s]'
parameter            'Period'                               unit '[s]'
parameter            'Period'                               unit '[s]'
parameter            'Period'                               unit '[s]'
parameter            'Period'                               unit '[s]'
parameter            'Direction'                            unit '[N^o]'
parameter            'Direction'                            unit '[N^o]'
parameter            'Direction'                            unit '[N^o]'
parameter            'Direction'                            unit '[N^o]'
parameter            'Direction'                            unit '[N^o]'
parameter            'Direction'                            unit '[N^o]'
parameter            'DirSpreading'                         unit '[-]'
parameter            'DirSpreading'                         unit '[-]'
parameter            'DirSpreading'                         unit '[-]'
parameter            'DirSpreading'                         unit '[-]'
parameter            'DirSpreading'                         unit '[-]'
parameter            'DirSpreading'                         unit '[-]'
0.00   5.5300 1.8600 1.8600 1.9100 1.8400 1.7100...
       8.2400 8.2400 8.2400 8.2400 8.4700 8.4700...
       -171.0700 -173.5300 -173.5300 -167.2300 -160.5600 -154.3000...
       2.0000 2.0000 2.0000 2.0000 2.0000 2.0000
60.00  3.5300 3.8600 1.8600 3.9100 3.8400 3.7100...
       8.2400 8.2400 8.2400 8.2400 8.4700 8.4700...
       -171.0700 -173.5300 -173.5300 -167.2300 -160.5600 -154.3000...
       2.0000 2.0000 2.0000 2.0000 2.0000 2.0000
location             'Boundary South     '
time-function        'non-equidistant'
reference-time       20060105
time-unit            'minutes'
interpolation        'linear'
parameter            'time               '                  unit '[min]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'WaveHeight'                           unit '[m]'
parameter            'Period'                               unit '[s]'
parameter            'Period'                               unit '[s]'
parameter            'Period'                               unit '[s]'
parameter            'Period'                               unit '[s]'
parameter            'Period'                               unit '[s]'
```

DRAFT

```
parameter            'Period'                                  unit '[s]'
parameter            'Period'                                  unit '[s]'
parameter            'Period'                                  unit '[s]'
parameter            'Period'                                  unit '[s]'
parameter            'Direction'                               unit '[N^o]'
parameter            'Direction'                               unit '[N^o]'
parameter            'Direction'                               unit '[N^o]'
parameter            'Direction'                               unit '[N^o]'
parameter            'Direction'                               unit '[N^o]'
parameter            'Direction'                               unit '[N^o]'
parameter            'Direction'                               unit '[N^o]'
parameter            'Direction'                               unit '[N^o]'
parameter            'Direction'                               unit '[N^o]'
parameter            'DirSpreading'                            unit '[-]'
parameter            'DirSpreading'                            unit '[-]'
parameter            'DirSpreading'                            unit '[-]'
parameter            'DirSpreading'                            unit '[-]'
parameter            'DirSpreading'                            unit '[-]'
parameter            'DirSpreading'                            unit '[-]'
parameter            'DirSpreading'                            unit '[-]'
parameter            'DirSpreading'                            unit '[-]'
parameter            'DirSpreading'                            unit '[-]'
0.00  1.2700 1.2700 1.2700 1.2700 1.3600 1.6000 1.3400 3.3400 3.0500...
      8.4700 8.4700 8.4700 8.4700 8.1600 7.3500 7.1200 7.1200 7.0800...
      -147.8800 -147.8800 -147.8800 -147.8800 -178.7700  173.9500  175.0400  175.0400 -179.1200...
      2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000
60.00 3.2700 1.2700 1.2700 3.2700 3.3600 3.6000 3.3400 3.3400 3.0500...
    8.4700 8.4700 8.4700 8.4700 8.1600 7.3500 7.1200 7.1200 7.0800...
    147.8800 -147.8800 -147.8800 -147.8800 -178.7700  173.9500  175.0400  175.0400 -179.1200...
    2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000
```

Example 3

If one would like to have a wave model with space-varying wave boundary conditions, with time-varying but spatial uniform wind field, one should add them to Datagroup *General* as follows:

```
[WaveFileInformation]
   FileVersion          = 02.00
[General]
   ProjectName          = Carrara
   ProjectNr            = 001
   Description          =
   Description          = Carrara test run
   OnlyInputVerify      = false
   SimMode              = stationary
   DirConvention        = nautical
   ReferenceDate        = 2006-01-05
   TSeriesFile          = timeseries.bcw
...
```

In Datagroup *TimePoint* the following should be added:

```
...
[TimePoint]
   Time                 =  6.0000000e+001
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
   WindSpeed            =  20.0
   WindDir              =  20.0
[TimePoint]
   Time                 =  1.2000000e+002
```

```
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
   WindSpeed            =  15.0
   WindDir              =  15.0
[TimePoint]
   Time                 =  1.8000000e+002
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
   WindSpeed            =  10.0
   WindDir              =  10.0
[TimePoint]
   Time                 =  2.4000000e+002
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
   WindSpeed            =  2.0
   WindDir              =  2.0
...
```

In Datagroup *Boundary* the following should be added:

```
...
[Boundary]
   Name                 = Boundary West
   Definition           = xy-coordinates
   StartCoordX          =  5.0000000e+005
   EndCoordX            =  5.0000000e+005
   StartCoordY          =  4.9274090e+006
   EndCoordY            =  4.7885805e+006
   SpectrumSpec         = parametric
   SpShapeType          = jonswap
   PeriodType           = peak
   DirSpreadType        = power
   PeakEnhanceFac       =  3.3000000e+000
   GaussSpread          =  9.9999998e-003
   CondSpecAtDist       =  2.7765670e+004
   CondSpecAtDist       =  5.5531340e+004
   CondSpecAtDist       =  6.3297008e+004
   CondSpecAtDist       =  8.3297008e+004
   CondSpecAtDist       =  1.1106268e+005
   CondSpecAtDist       =  1.3882834e+005
[Boundary]
   Name                 = Boundary South
   Definition           = xy-coordinates
   StartCoordX          =  5.0000000e+005
   EndCoordX            =  6.2226400e+005
   StartCoordY          =  4.7608150e+006
   EndCoordY            =  4.7608150e+006
   SpectrumSpec         = parametric
   SpShapeType          = jonswap
   PeriodType           = peak
   DirSpreadType        = power
   PeakEnhanceFac       =  3.3000000e+000
   GaussSpread          =  9.9999998e-003
   CondSpecAtDist       =  0.0000000e+000
   CondSpecAtDist       =  1.0000000e+003
   CondSpecAtDist       =  1.0000000e+004
   CondSpecAtDist       =  2.0377330e+004
   CondSpecAtDist       =  4.0754660e+004
   CondSpecAtDist       =  6.1131988e+004
   CondSpecAtDist       =  8.1509320e+004
   CondSpecAtDist       =  1.0188665e+005
   CondSpecAtDist       =  1.2226398e+005
```

DRAFT

...

The <bcw>-file, which is defined in section A.2.3, should be the same as that in Example 2.

A.2.8.3 Space-varying wave boudnary conditions using for UNIBEST coupling (<md-vwac>-file)

For the coastline model UNIBEST, wave computations can be required representing a wave climate. Such a wave climate is schematized into several wave conditions and corresponding wind conditions. These wave and wind conditions can be defined all in one file: the so-called <md-vwac>-file. This file must be added to the working directory of the wave model. Only when this file is present in the working directory, wave computations will be carried out for all wave conditions in the <md-vwac>-file. In this way a large number of wave conditions can be computed in a batch mode.

File contents:	List of wave and wind conditions for UNIBEST model with no time points
File type:	free formatted/unformatted.
Restrictions:	maximum record length in the (free) formatted file is 132.
Example:	formatted file of a <md-vwac.*runid*>

```
* Name of main SCO file:  NZ_STORM.SCO
UNIBEST *(MORSYS/UNIBEST)
10 *total number of wave conditions
* Hm0   Tp    theta   ms   H0    U10     theta_wind
* (m)   (s)   (N°)    -    (m)   (m/s)   (N°)
1.0     5     330     4    0.2   0       0
1.5     5     310     4    0.1   0       0
3.0     8     350     4    0.4   0       0
2.2     7     270     4    0.3   0       0
```

Description of parameters:

H_{m0} [m]	Significant wave height in metres; this value will be prescribed on all specified wave boundaries.
T_p [s]	Peak period of the energy spectrum. This value will be prescribed on all specified wave boundaries.
theta [N°]	Mean wave direction according to the Nautical or Cartesian convention (in degrees). This value will be prescribed on all specified wave boundaries.
ms [-]	Width energy distribution. This is the directional standard deviation in degrees if the option *Degrees* is chosen in the sub-window *Spectral space* or it is the power m if the option *Cosine power* is chosen in the same above sub-window.
H0 [m]	The additional water level over the entire wave model. The water level is measured positively upward from the same datum from which the bottom levels are taken.
U10 [m/s]	Wind velocity at 10 m elevation.
Theta_wind [N°]	Wind direction at 10 m elevation according to the convention, specified in the sub-window **Constants**.

Remarks:

- ◇ On the third line of the md-vwac file the amount of wave conditions is given. In the mdw-file or in the WAVE-GUI an equal amount of time points must be prescribed matching with the amount of wave conditions in the md-vwac file.
- ◇ The defined wave boundary conditions are overruled by the prescribed wave conditions

DRAFT

in the md-vwac file.

A.2.8.4 Time- and space-varying wave boundary conditions: TPAR file

TPAR files containing non-stationary wave parameters. A TPAR file is for only one section of the boundaries. For space-varying, the user has to define multiple TPAR files. The TPAR file has the string TPAR on the first line of the file and a number of lines which each contain 5 numbers:

1 Time (ISO notation),
2 Hs,
3 Period (average or peak period depending on the choice given in the Swan Spectral Space under *Edit Spectral space*),
4 Peak Direction (Nautical or Cartesian, depending on the settings in the *Physical parameters*),
5 Directional spread (in degrees or as power of Cos depending on the choice given in the Swan Spectral Space under *Edit Spectral space*).

Example of a TPAR file (for example, the filename is TPAR01.bnd):

```
TPAR
19920516.1300 4.2 12.  -110. 22.
19920516.1800 4.2 12.  -110. 22.
19920517.0000 1.2  8.  -110. 22.
19920517.1200 1.4  8.5  -80. 26.
19920517.2000 0.9  6.5  -95. 28.
```

Thus in the mdw file, the corresponding segment is:

```
...
[Boundary]
   Name                = Bound1
   Definition          = grid-coordinates
   StartCoordM         = 0
   EndCoordM           = 0
   StartCoordN         = 0
   EndCoordN           = 39
   SpectrumSpec        = from file
   Spectrum            = TPAR01.bnd
...
```

The boundary section is defined in MN format.

A.2.9 Spectral input and output files

There are two types of Spectrum files:

◇ files containing stationary or non-stationary 1D spectra (usually from measurements)
◇ files containing stationary or non-stationary 2D spectra (from other computer programs or other SWAN runs).

The structure of the files containing 1D or 2D spectra is described below (there is no relation with the definition of the boundary file generated by WAM or WAVEWATCH III). 1D and 2D files can be used for one or more than one location. The spectral frequencies (and directions in the case of a 2D spectrum) do not have to coincide with the frequencies and directions used

DRAFT

in the present WAVE (SWAN) run (in a nested run SWAN will interpolate to these frequencies and directions). The co-ordinates of locations in the 1D and 2D files are ignored when SWAN reads this.

This appendix describes the format of the files for spectral input (command BOUNDARY) and output (commands SPEC and NEST) by SWAN. The files are recognised by SWAN or another reading program by the presence of the keyword SWAN and a version number on the first line of the file. This description is valid for version number 1.

These files contain the following information:

- ◇ co-ordinates of locations
- ◇ frequencies
- ◇ directions (if used for 2D)
- ◇ time (if time-dependent)
- ◇ spectral energy or variance densities (and aver. dir. and dir. spread if 1D)

Example of a 1D non-stationary spherical co-ordinates file:

```
SWAN 1                          Swan standard spectral file, version
$ Data produced by SWAN version 40.41
$ Project:'projname' ;          run number:          'runnum'
TIME                            time-dependent data
1                               time coding option
LONLAT                          locations in spherical co-ordinates
2                               number of locations
1.00  1.00
1.20  1.00
RFREQ                           relative frequencies in Hz
25                              number of frequencies
0.0418
0.0477
0.0545
0.0622
0.0710
0.0810
0.0924
0.1055
0.1204
0.1375
0.1569
0.1791
0.2045
0.2334
0.2664
0.3040
0.3470
0.3961
0.4522
0.5161
0.5891
0.6724
0.7675
0.8761
1.0000
QUANT
3                               number of quantities in table
VaDens                          variance densities in m2/Hz
m2/Hz                           unit
-0.9900E+02                     exception value
CDIR                            average Cartesian direction in degr
```

DRAFT

```
degr                          unit
-0.9990E+03                   exception value
DSPRDEGR                      directional spreading
degr                          unit
-0.9000E+01                   exception value
19680606.030000               date and time
LOCATION 1
0.3772E-03  190.1   6.3
0.1039E-02  190.2   6.5
0.2281E-02  190.3   6.7
0.3812E-02  190.3   6.7
0.4255E-02  190.3   6.6
0.2867E-02  190.1   6.3
0.1177E-02  189.6   5.8
0.3892E-03  192.0  15.2
0.8007E-03  244.5  22.9
0.6016E-02  251.4  11.5
0.1990E-01  251.0  11.0
0.3698E-01  249.9  10.9
0.3874E-01  248.1  12.1
0.2704E-01  246.6  13.0
0.1672E-01  247.0  13.5
0.1066E-01  247.7  13.7
0.5939E-02  247.3  14.0
0.3247E-02  246.5  14.6
0.1697E-02  245.9  14.9
0.8803E-03  245.6  15.1
0.4541E-03  245.5  15.3
0.2339E-03  245.4  15.5
0.1197E-03  245.5  15.6
0.6129E-04  245.5  15.7
0.3062E-04  245.3  15.9
LOCATION 2
0.7129E-02   67.2  25.3
0.3503E-01   67.5  21.7
0.1299E+00   68.2  19.7
0.5623E+00   69.7  18.0
0.1521E+01   71.4  18.0
0.3289E+01   74.0  18.8
0.4983E+01   77.2  20.3
0.4747E+01   79.9  22.0
0.2322E+01   79.4  30.7
0.1899E+01  341.1  56.2
0.1900E+01  314.6  39.4
0.6038E+01  324.3  31.9
0.8575E+01  326.1  31.0
0.4155E+01  325.1  30.5
0.1109E+01  322.8  32.9
0.7494E+00  323.1  33.3
0.4937E+00  323.1  33.3
0.2953E+00  323.3  33.7
0.1661E+00  323.6  34.0
0.9788E-01  323.7  33.8
0.5766E-01  323.8  33.6
0.3397E-01  324.0  33.5
0.2001E-01  324.1  33.4
0.1179E-01  324.2  33.3
0.6944E-02  324.2  33.2
```

Example of a 2D stationary Cartesian co-ordinates file:

```
SWAN 1                        Swan standard spectral file, version
$ Data produced by SWAN version 40.41
$ Project:'projname' ;        run number:'runnum'
```

DRAFT

```
LOCATIONS                     locations in x-y-space
2                             number of locations
0.00  0.00
22222.22  0.00
RFREQ                         relative frequencies in Hz
25                            number of frequencies
0.0418
0.0477
0.0545
0.0622
0.0710
0.0810
0.0924
0.1055
0.1204
0.1375
0.1569
0.1791
0.2045
0.2334
0.2664
0.3040
0.3470
0.3961
0.4522
0.5161
0.5891
0.6724
0.7675
0.8761
1.0000
CDIR                          spectral Cartesian directions in degr
24                            number of directions
7.5000
22.5000
37.5000
52.5000
67.5000
82.5000
97.5000
112.5000
127.5000
142.5000
157.5000
172.5000
187.5000
202.5000
217.5000
232.5000
247.5000
262.5000
277.5000
292.5000
307.5000
322.5000
337.5000
352.5000
QUANT
1                             number of quantities in table
VaDens                        variance densities in m2/Hz/degr
m2/Hz/degr                    unit
-0.9900E+02                   exception value
FACTOR
0.422574E-11
0  0  0  0  0  0  0  0  0  0  0  0     1    0  0
0  0  0  0  0  0  0  0  0
```

DRAFT

```
0  0  0  0  0  0  0  0  0  0  0  0    44    3  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0   817   60  1
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  8018  574  9
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0 39230 2532 38
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0 92174 4477 68
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0 99010 1946 29
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0 47054  131  2
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0 13228    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0 39417    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0 61269    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0 29738    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  2161    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     2    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0     0    0  0
0  0  0  0  0  0  0  0  0
FACTOR
0.675611E-06
51 242 574 956 1288 1482 1481 1286 957 579 244 51 0 0 0
0 0 0 0 0 0 0 0 0
129 610 1443 2402 3238 3725 3724 3234 2406 1454 613 128 0 0 0
0 0 0 0 0 0 0 0 0
273 1287 3054 5084 6846 7872 7869 6837 5091 3076 1295 271 0 0 0
0 0 0 0 0 0 0 0 0
665 3152 7463 12402 16712 19229 19221 16690 12419 7518 3172 662 0 0 0
0 0 0 0 0 0 0 0 0
1302 6159 14608 24275 32688 37618 37603 32644 24309 14716 6198 1296 0 0 0
0 0 0 0 0 0 0 0 0
2328 10989 26020 43341 58358 67109 67080 58281 43401 26213 11058 2317 0 0 0
0 0 0 0 0 0 0 0 1
3365 15922 37712 62733 84492 97150 97110 84380 62820 37991 16021 3349 0 0 0
0 0 0 0 0 0 0 0 1
3426 16230 38440 63939 86109 99010 98969 85995 64027 38724 16331 3410 0 0 0
0 0 0 0 0 0 0 0 0
```

DRAFT

```
2027 9612 22730 37790 50909 58529 58505 50841 37843 22898 9672 2018 0 0 0
0 0 0 0 0 0 0 0 0
672 3178 7538 12535 16892 19440 19432 16870 12552 7594 3198 669 0 0 0
0 0 0 0 0 0 0 0 0
101 479 1135 1890 2542 2924 2923 2539 1892 1144 482 101 0 0 0
0 0 0 0 0 0 0 0 0
2 11 26 43 57 66 66 57 43 26 11 2 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

Note that the true variance or energy densities are obtained by multiplying each number with the factor given under the keyword FACTOR.

A.2.10 Space-varying wind field

This feature has been made available as a special feature in Delft3D-WAVE. It can not (yet) be switched on in the WAVE-GUI. The user can include this functionality by adding the keyword `Meteofile` in the MDW-file. The keyword should specify the file containing the space-varying wind data. If one wishes to specify wind fields that vary in space but are constant in time, one should simply incorporate the same wind field data block twice in one file. This generates a wind field that is constant in time.

Remarks:

- ◇ The keyword `Meteofile` can be added both in Datagroup *General* as in Datagroup *Domain*. When the keyword is added in Datagroup *General*, the wind will be incorporated in all domains. When the keyword is added in Datagroup *Domain*, the wind will be incorporated in that domain only.
- ◇ The `Meteofile` may occur more than once in the MDW-file to specify multiple sets of meteorological data (also within a Datagroup).

Example 1

If one would like to add two meteofiles containing an x-component and y-component for space-varying wind, respectively, and apply the wind to all domains of the WAVE simulation, one should add them to Datagroup *General* as follows:

```
[WaveFileInformation]
   FileVersion          = 02.00
[General]
   ProjectName          = Siu-Lam
   ProjectNr            = 001
   Description          = Tutorial Delft3D-WAVE
   Description          = Siu Lam model
   Description          = SWAN wave model using a curvilinear grid
   OnlyInputVerify      = false
   SimMode              = quasi-stationary
   DirConvention        = nautical
   ReferenceDate        = 2005-10-01
   ObstacleFile         = obst_data_keyw.obs
   MeteoFile            = xwind.wnd
   MeteoFile            = ywind.wnd
[TimePoint]
   ....
```

Example 2

If one would like to add the same meteorological files, but apply them only in the domain with grid *siu_lam_coarse.grd*, one should add them to Datagroup *Domain* as:

```
[WaveFileInformation]
   FileVersion          = 02.00
[General]
   ProjectName          = Siu-Lam
   ProjectNr            = 002
   Description          = Tutorial Delft3D-WAVE
   Description          = Siu Lam model, 2 domains
   Description          = SWAN wave model using 2 curvilinear grids
   OnlyInputVerify      = false
   SimMode              = quasi-stationary
   DirConvention        = nautical
   ReferenceDate        = 2005-10-01
   ObstacleFile         = obst_data_keyw.obs
[TimePoint]
   ....
[Domain]
   Grid                 = siu_lam_coarse.grd
   BedLevel             = siu_lam_coarse.dep
   DirSpace             = circle
   NDir                 = 36
   StartDir             =  0.00000000000000000e+000
   EndDir               =  0.00000000000000000e+000
   FreqMin              =  5.00000074505806000e-002
   FreqMax              =  1.00000000000000000e+000
   NFreq                = 24
   Output               = true
   MeteoFile            = xwind.wnd
   MeteoFile            = ywind.wnd
[Domain]
   Grid                 = siu_lam_fine.grd
   BedLevel             = siu_lam_fine.dep
   DirSpace             = circle
   NDir                 = 36
   StartDir             =  0.00000000000000000e+000
   EndDir               =  0.00000000000000000e+000
   FreqMin              =  5.00000074505806000e-002
   FreqMax              =  1.00000000000000000e+000
   NFreq                = 24
   Output               = true
[Boundary]
   ....
```

DRAFT

Remark:

- ◇ When applying space-varying wind in only one or some of the domains, the user should be aware of the fact that the transition in wind forcing from one domain to the other may be not smooth.

In many cases the space varying wind data is provided by a meteorological station. This data is often defined on a different grid than the computational grid used in Delft3D-WAVE. Translating these files into files defined on the (curvilinear) grid of the computational engine is often a lengthy process and can result in huge files. This special feature facilitates the reading of the meteorological data on its own grid and interpolates the data internally to the grid of Delft3D-WAVE.

Delft3D-WAVE can handle wind data on several different types of grids:

1 Space-varying wind on the computational (SWAN) grid
2 Space-varying wind on an equistant grid
3 Space-varying wind on a curvilinear grid
4 Space-varying wind on a Spiderweb grid

For these types of meteorological input, fixed formats have been set-up, that completely define a dataset. This form of meteorological input is also used by Delft3D-FLOW, see (Delft3D-FLOW, 2013). In Delft3D-FLOW, also the atmospheric pressure is read from the meteorological files and used in the simulation. This is not (yet) available in Delft3D-WAVE. In the following sections, generic descriptions of the formats of the meteorological input types are given. In these descriptions the atmospheric pressure is also considered. This is not relevant for Delft3D-WAVE and may be excluded. For *Space-varying wind on the computational (SWAN) grid*, both `x_wind`, `y_wind` and `air_pressure` are given in one file. Similarly, for *Space-varying wind on a Spiderweb grid*, both `wind_speed`, `wind_from_direction` and `p_drop` (atmospheric pressure drop) are specified in one file. This format must also be used for a Delft3D-WAVE simulation, for which the atmospheric pressure (drop) is then not used.

A.2.10.1 Space-varying wind on the computational (SWAN) grid

File contents	Time-series for space varying wind velocity components (east-west and south-north) and atmospheric pressure, defined on the computational grid. The file consists of a header, followed by datablocks containing the wind and pressure fields at times specified using a standardised time definition above each datablock. The header specifies the type of file and the input it contains using a number of keywords. The keywords are case insensitive and the order of the keywords is not fixed.
Filetype	ASCII or binary.
File format	Free formatted or unformatted, keyword based.
Filename	<*name*.wnd>
Generated	Some offline program.

Header description:

Keywords	Value	Description
FileVersion	1.03	version of file format
Filetype	meteo_on_computational_grid	meteo input on computational grid
NODATA_value	*free*	value used for input that is to be neglected
n_quantity	3	number of quantities specified in the file
quantity1	x_wind	wind in x-direction
quantity2	y_wind	wind in y-direction
quantity3	air_pressure	air pressure
unit1	m s-1	unit of quantity1, meters/second
unit2	m s-1	unit of quantity2, meter/second
unit3	Pa *or* mbar	unit of quantity3, Pa or millibar

Time definition and data block description

Keywords	Value	Description
Time	*fixed format described below*	time definition string

The time definition string has a fixed format, used to completely determine the time at which a dataset is valid. The time definition string has the following format:

TIME *minutes/hours since* YYYY-MM-DD HH:MM:SS TIME ZONE, e.g.

```
360 minutes since 2008-07-28 10:55:00 +01:00
```

The format of the string is completely fixed. No extra spaces or tabs can be added between the different parts of the definition. The time definition is followed by the datablock of input values corresponding to the specified time. The data block consists of three subsequent blocks containing the velocity component in M-direction, the velocity component in N-direction and the atmospheric pressure, respectively. All three quantities are given for *Nmax* by *Mmax* points, where the first value in the dataset corresponds to cell $(1, 1)$ on the grid. Every next line in the dataset then corresponds to a row on the grid. The time definition and the data block — for all three quantities — are repeated for each time instance of the time-series.

DRAFT

File version and conversion

The current description holds for `FileVersion` 1.03. The table below shows the latest modifications in the file format (and version number).

FileVersion	Modifications
`1.03`	No changes for this meteo input type, but for the meteo types *meteo_on_equidistant_grid* and *meteo_on_curvilinear_grid*
`1.02`	No changes for this meteo input type, but for the meteo type *meteo_on_spider_web_grid*
`1.01`	Changed keyword `MeteoType` to `FileType` Changed fixed value of input type (Keyword `Filetype`) from *Svwp* to *meteo_on_computational_grid* (*meteo_on_flow_grid* is also allowed)

Restrictions:

- ◇ Keywords are followed by an equal sign '=' and the value of the keyword.
- ◇ When a keyword has value *free* the value of this keyword is free to choose by the user. When only one value is given for a keyword, this keyword has a fixed value and when 2 or more options are shown, the user can choose between those values.
- ◇ Times must be specified exactly according to the time definition. See the examples shown in this section.
- ◇ The contents of the file will not be checked on its domain.
- ◇ The wind components are specified at the cell centres (water level points) of the computational grid.
- ◇ Input items in a data block are separated by one or more blanks (free formatted file only).

Remarks:

- ◇ The time definition in the meteorological file contains the number of minutes or hours since a reference data and time in a certain time zone. The reference time and time zone may differ from those of the simulation. The computational engine will search in the meteo file for the simulation time and interpolate between neighbouring times if necessary. Possible differences in time zone will be accounted for by shifting the meteo input data with the difference. The reference times within the time definition string may vary in a meteo file, i.e. it is possible to attach new input with a different reference time, behind the last data block.
- ◇ Comments can be added after #'s.

Example

Model area of $25 * 33$ grid points (Mmax = 25; Nmax = 33). The input data is printed in Courier, comments are printed behind #'s.

DRAFT

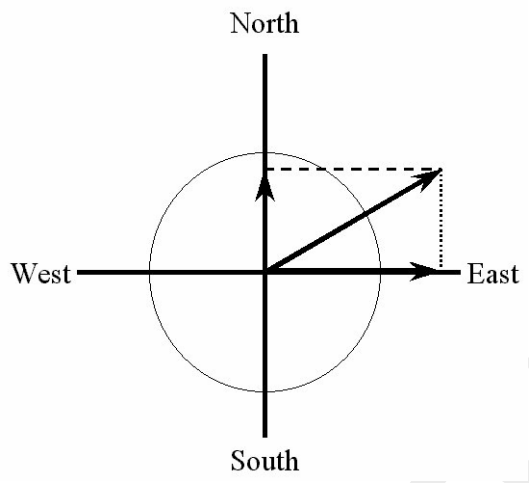

Figure A.1: *Definition wind components for space varying wind*

```
Time = 0.0 minutes since 2008-09-20 10:30:00 +01:00      # Time definition
{33 records with 25 values each}                         # Wind component west to east
{33 records with 25 values each}                         # Wind component south to north
{33 records with 25 values each}                         # Atmospheric pressure
Time = 340.0 minutes since 2008-09-20 10:30:00 +01:00    # Time definition
{33 records with 25 values each}                         # Wind component west to east
{33 records with 25 values each}                         # Wind component south to north
{33 records with 25 values each}                         # Atmospheric pressure
Time = 600.0 minutes since 2008-09-20 10:30:00 +01:00    # Time definition
{33 records with 25 values each}                         # Wind component west to east
{33 records with 25 values each}                         # Wind component south to north
{33 records with 25 values each}                         # Atmospheric pressure
Time = 1240.0 minutes since 2008-09-20 10:30:00 +01:00   # Time definition
{33 records with 25 values each}                         # Wind component west to east
{33 records with 25 values each}                         # Wind component south to north
{33 records with 25 values each}                         # Atmospheric pressure
```

Remarks:

- ◇ To obtain the wind direction according to the nautical convention, the wind direction is reversed.
- ◇ The wind is specified in terms of its components along the west-east (`x_wind`) and south-north (`y_wind`) co-ordinate system, see Figure A.1. These definitions differ from the nautical convention (used for uniform wind), which is specified relative to true North, see Figure A.2.

A.2.10.2 Space-varying wind on an equistant grid

File contents	Time-series of a space varying wind and atmospheric pressure defined on an equidistant (Cartesian or spherical) grid.
File format	Free formatted or unformatted, keyword based.
Generated	Some offline program.

Remark:

- ◇ The keywords are case insensitive.

DRAFT

Keywords	Value	Description

Header description for the wind velocity files:

Keywords	Value	Description
FileVersion	1.03	version of file format
Filetype	meteo_on_equidistant_grid	meteo input on equidistant grid
NODATA_value	*free*	value used for input that is to be neglected
n_cols	*free*	number of columns used for wind datafield
n_rows	*free*	number of rows used for wind datafield
grid_unit	m *or* degree	unit of distances on the grid in both x- and y-direction
x_llcorner	*free*	x-coordinate of lower left corner of lower left grid cell (in units specified in grid_unit
y_llcorner	*free*	y-coordinate of lower left corner of lower left grid cell (in units specified in grid_unit
x_llcenter	*free*	x-coordinate of centre of lower left grid cell (in units specified in grid_unit
y_llcenter	*free*	y-coordinate of centre of lower left grid cell (in units specified in grid_unit
dx	*free*	gridsize in x-direction in units specified in grid_unit
dy	*free*	gridsize in y-direction in units specified in grid_unit
n_quantity	1	number of quantities specified in the file
quantity1	x_wind *or* y_wind	the velocity component given in unit unit1
unit1	m s-1	unit of quantity1: metre/second

The user must specify the location of the equidistant grid on which the meteorological data is specified. If one has the location of the lower left corner of the lower left grid cell, one can spec-

DRAFT

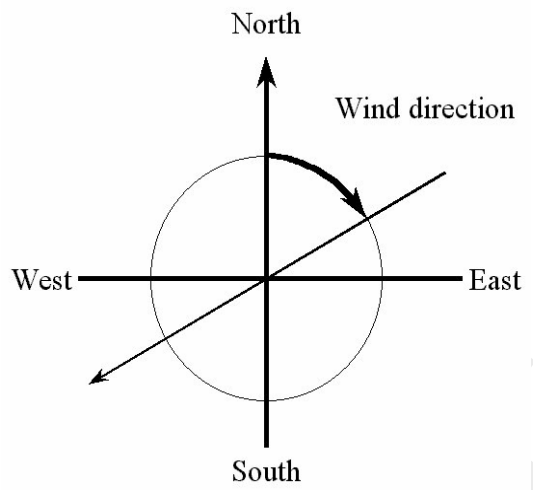

Figure A.2: *Definition sketch of wind direction according to Nautical convention*

ify the starting point of the grid using keywords `x_llcorner` and `y_llcorner`. If one has the location of the cell centre of the lower left grid cell, one should use the keywords `x_llcenter` and `y_llcenter`. Using the first option, the first data value is placed at (x_llcorner+$\frac{1}{2}dx$, y_llcorner+$\frac{1}{2}dy$), which is the cell centre of cell (1,1). Using the latter option, the first data value is placed at (x_llcenter, y_llcenter), which is again the cell centre of cell (1,1), i.e. the data values are always placed at the cell centres of the meteorological grid. Note that the lower left grid cell is defined to be the grid cell with index (1,1). When using the option of meteorological data on a separate curvilinear grid, the origin and orientation of the data set can be chosen freely with respect to the grid on which it is specified, see section A.2.10.3 for details.

Time definition and data block description for the wind velocity files

Keywords	**Value**	**Description**
`Time`	*fixed format described below*	time definition string

The time definition string has a fixed format, used to completely determine the time at which a dataset is valid. The time definition string has the following format:

TIME *minutes/hours since* YYYY-MM-DD HH:MM:SS TIME ZONE, e.g.

```
360 minutes since 2008-07-28 10:55:00 +01:00
```

The format of the string is completely fixed. No extra spaces or tabs can be added between the different parts of the definition. The time definition is followed by the datablock of input values corresponding to the specified time. The data block contains values for the wind velocity in x- or y-direction for *n_cols* by *n_rows* points, starting at the top left point. The time definition and the data block are repeated for each time instance of the time-series.

The atmospheric pressure file

The header for the atmospheric pressure is similar to that of the wind velocity files, except for the following differences.

Keywords	**Value**	**Description**
`quantity1`	air_pressure	air pressure
`unit1`	Pa *or* mbar	unit of `quantity1`: Pascal or

File version and conversion

The current description holds for `FileVersion` 1.03. The table below shows the latest modifications in the file format (and version number).

FileVersion	Modifications
`1.03`	Use of keyword `Value_pos` to indicate the position of the lower left corner of the grid replaced by use of the combination of keywords: `x_llcorner` and `y_llcorner` or `x_llcenter` and `y_llcenter`
`1.02`	No changes for this meteo input type, but for the meteo type *meteo_on_spiderweb_grid*
`1.01`	Changed keyword `MeteoType` to `FileType` Changed fixed value of input type (Keyword `Filetype`) from *ArcInfo* to *meteo_on_equidistant_grid*

Restrictions:

- ◇ The contents of the file will not be checked on its domain.
- ◇ Keywords are followed by an equal sign '=' and the value of the keyword.
- ◇ When a keyword has value *free*, the value of this keyword is free to choose by the user. When only one value is given for a keyword, this keyword has a fixed value and when 2 or more options are shown, the user can choose between those values.
- ◇ Times must be specified exactly according to the time definition. See the examples shown in this section.
- ◇ The atmospheric pressure file must use the same grid definition and time frame as the files for the wind velocity components.
- ◇ The unit of the meteo grid must be the same as the computational grid, i.e. both with `grid_unit` = [m] or both with `grid_unit` = [degree].
- ◇ Input items in a data block are separated by one or more blanks.
- ◇ The wind components are specified at the cell centres (water level points) of the numerical grid.
- ◇ The wind components are specified in the west-east (`x_wind`) and south-north directions (`y_wind`).

Remarks:

- ◇ The time definition in the meteo files contains the number of minutes or hours since a reference date and time in a certain time zone. The reference time and time zone may differ from those of the simulation. During a simulation the computational engine will search in the meteo file for the current simulation time and interpolate between neighbouring times if necessary. Possible differences in time zone will be accounted for by shifting the meteo input data. The reference times within the time definition string may vary in a meteo file, i.e. it is possible to attach new input with a different reference time, behind the last data block. Consecutive times must always be increasing in the input file.
- ◇ Comments can be added after pound signs (#). These are not read.

Example of a file containing wind in x-direction (west-east)

The data blocks in this example are the result of the following FORTRAN statements:

```
do j = nrows,1,-1
```

DRAFT

```
   write(out,*) (xwind(i,j),i=1,ncols)
enddo
```

The x-wind velocity file for a 3 (n_cols) by 4 (n_rows) grid has the following layout:

```
FileVersion      =    1.03
filetype         =    meteo_on_equidistant_grid
NODATA_value     =    -999.000
n_cols           =    3
n_rows           =    4
grid_unit        =    degree
x_llcenter       =    -12.000
y_llcenter       =     48.000
dx               =    0.12500
dy               =    0.083333333
n_quantity       =    1
quantity1        =    x_wind
unit1            =    m s-1
TIME =      0.0 hours since 2008-01-15 04:35:00 +00:00
2   3.0  3.6
3   4.5  2
2.2 1    2.3
1.2 0.7 -0.4
TIME =      6.0 hours since 2008-01-15 04:35:00 +00:00
-1.1 -2.3 -3.6
-3.2  0.8  1.1
 2.2 -1   -1.6
 1.2 -0.7 -0.4
```

This results in an x-component of wind velocity given - in [m/s] - on a spherical, 3 by 4, equidistant grid, with grid sizes given by *dx* and *dy* (in degrees) and where the centre point of the lower left cell of the grid lies in (longitude, latitude) (-12.0, 48.0) on the globe. Data is given at two times: 0 and 6 hours since January 15th, 2008, 4:35 AM, in UTC+0.

A.2.10.3 Space-varying wind on a curvilinear grid

File contents	Time-series of a space varying wind and atmospheric pressure defined on a curvilinear (Cartesian or spherical) grid.
File format	Free formatted or unformatted, keyword based.
Generated	Some offline program.

Remark:

◇ The keywords are case insensitive.

Header description for the wind velocity files:

Keywords	Value	Description
`FileVersion`	1.03	version of file format
`Filetype`	meteo_on_curvilinear_grid	meteo input on curvilinear grid
`NODATA_value`	*free*	value used for input that is to be neglected
`grid_file`	*free*.grd	name of the curvilinear grid file on which the data is specified

DRAFT

Keywords	Value	Description
first_data_value	grid_llcorner *or* grid_ulcorner *or* grid_lrcorner *or* grid_urcorner	*see example below*
data_row	grid_row *or* grid_column	*see example below*
n_quantity	1	number of quantities specified in the file
quantity1	x_wind *or* y_wind	the velocity component given in unit unit1
unit1	m s-1	unit of quantity1: metres/second

Time definition and data block description for the wind velocity files

For a description of the time definition and data block see section A.2.10.2.

The atmospheric pressure file

For a description of the atmospheric file see section A.2.10.2.

File version and conversion

The current description holds for FileVersion 1.03. The table below shows the latest modifications in the file format (and version number).

FileVersion	Modifications
1.03	Fixed bug in interpolation of data from meteo grid to computational grid: Conversion script mirrored data set erroneously. This was treated correctly by meteo module. Fixed both the conversion script and the meteo module together: Required modification in meteo input file: Change first_data_value = grid_llcorner into grid_ulcorner or vice versa or Change first_data_value = grid_lrcorner into grid_urcorner or vice versa
1.02	No changes for this meteo input type, but for the meteo type *meteo_on_spiderweb_grid*
1.01	Changed keyword MeteoType to FileType Changed keyword Curvi_grid_file to Grid_file Changed fixed value of input type (Keyword Filetype) from *Curvi* to *meteo_on_curvilinear_grid*

DRAFT

Restrictions:

- ◇ The restrictions for space varying wind and pressure on a separate curvilinear grid are the same as for space varying wind and pressure on an equidistant grid, described in section A.2.10.2. A differerence is that the data values on the curvilinear grid are not specified in the cell centres, but in the grid points (cell corners).
- ◇ The unit of the meteo grid must be the same as the computational grid, i.e. both with `grid_unit` = [m] or both with `grid_unit` = [degree].

Remark:

- ◇ The remarks for space varying wind and pressure on a separate curvilinear grid are the same as for space varying wind and pressure on an equidistant grid, described in section A.2.10.2.

Example:

A file for input of x-velocity (in west-east direction) on a 4 by 5 curvilinear grid, where the meteorogical data is mirrored vertically with respect to the grid, has the following layout:

```
FileVersion      =    1.03
filetype         =    meteo_on_curvilinear_grid
NODATA_value     =    -999.000
grid_file        =    curviwind.grd
first_data_value =    grid_llcorner
data_row         =    grid_row
n_quantity       =    1
quantity1        =    x_wind
unit1            =    m s-1
TIME =      0.0 minutes since 1993-06-28 14:50:00 -02:00
 1   2   3   4   5
 6   7   8   9  10
11  12  13  14  15
16  17  18  19  20
TIME =      600.0 minutes since 1993-06-28 14:50:00 -02:00
 1   2   3   4   5
 6   7   8   9  10
11  12  13  14  15
16  17  18  19  20
```

This results in an x-component of velocity given - in [m/s] - on the curvilinear grid specified in file <curviwind.grd>. The data set will be mirrored such that the first value of the data (upper left corner, in the example '1') corresponds to the lower left corner of the grid (point (1,1)) and a row of data corresponds to a row on the grid, see Figure A.3. Data is given at two times: 0 and 600 minutes since June 28th, 1993, 14:50 PM, in UTC-2.

A.2.10.4 Space-varying wind on a Spiderweb grid

Cyclone winds are governed by a circular motion, combined with a cyclone track. This type of wind is generally very difficult to implement on a curvilinear grid. This feature facilitates the reading of the so-called Spiderweb files and interpolates the wind and pressure data internally to the computational grid. A special feature of the space varying wind and pressure on the Spiderweb grid is that it can be combined with one of the other meteorological input options described in this manual, i.e. to either uniform wind and pressure, or to one of the space varying wind and pressure options, see section A.2.10.

File contents	Time-series of a space varying wind and atmospheric pressure defined on a Spiderweb grid. This grid may be specified in Cartesian or spherical coordinates.

DRAFT

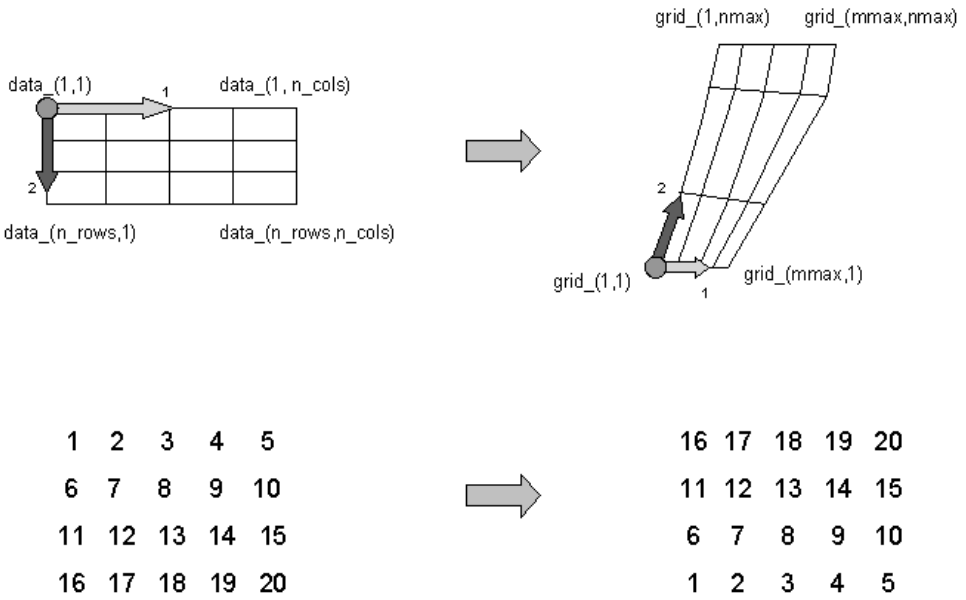

Figure A.3: *Illustration of the data to grid conversion for meteo input on a separate curvilinear grid*

File format	Free formatted or unformatted, keyword based.
Generated	Some offline program.

Remarks:

- ◇ The keywords are case insensitive.
- ◇ Space varying wind and pressure on a Spiderweb grid is added to other wind input and the wind fields are interpolated and combined in and around the cyclone.

Header description of the Spiderweb wind and pressure file:

Keywords	**Value**	**Description**
`FileVersion`	1.03	version of file format
`Filetype`	meteo_on_spiderweb_grid	meteo input on Spiderweb grid
`NODATA_value`	*free*	value used for input that is to be neglected
`n_cols`	*free*	number of gridpoints in angular direction
`n_rows`	*free*	number of gridpoints in radial direction
`grid_unit`	m *or* degree	unit of the Spiderweb grid
`spw_radius`	*free*	radius of the spiderweb given in units given by `spw_rad_unit`
`spw_rad_unit`	m	unit of the Spiderweb radius

DRAFT

Keywords	Value	Description
spw_merge_frac	[0.0,1.0]	fraction of the Spiderweb radius where merging starts of the background wind with the Spiderweb wind. Default is 0.5
air_pressure _reference	air_pressure_default_from _computational_engine *or free*	Both keyword and value are too long to fit on one line. Reference value related to p_drop is the default air pressure of the computional engine *or* the value specified. If missing, p_drop is extracted from the actual atmospheric pressure.
n_quantity	3	number of quantities specified in the file
quantity1	wind_speed	wind speed given in unit unit1
quantity2	wind_from_direction	direction where the wind is coming from given in unit unit2
quantity3	p_drop	drop in atmospheric pressure given in unit unit3
unit1	m s-1	unit of quantity1: metres/second
unit2	degree	unit of quantity2: degrees
unit3	Pa *or* mbar	unit of quantity3: Pascal or millibar

Time definition and data block description

For a description of the time definition see section A.2.10.2.

Cyclone track information:

For each time in the time series of space varying wind and pressure on a Spiderweb grid, the position of the cyclone eye (and thus also the spiderweb grid) must be given, as well as the drop of atmospheric pressure in the cyclone eye.

DRAFT

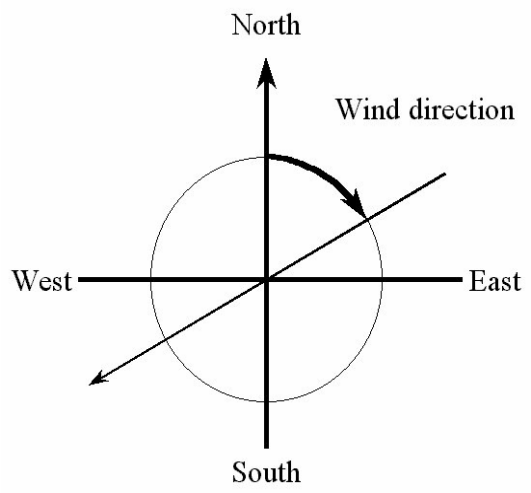

Figure A.4: *Wind definition according to Nautical convention*

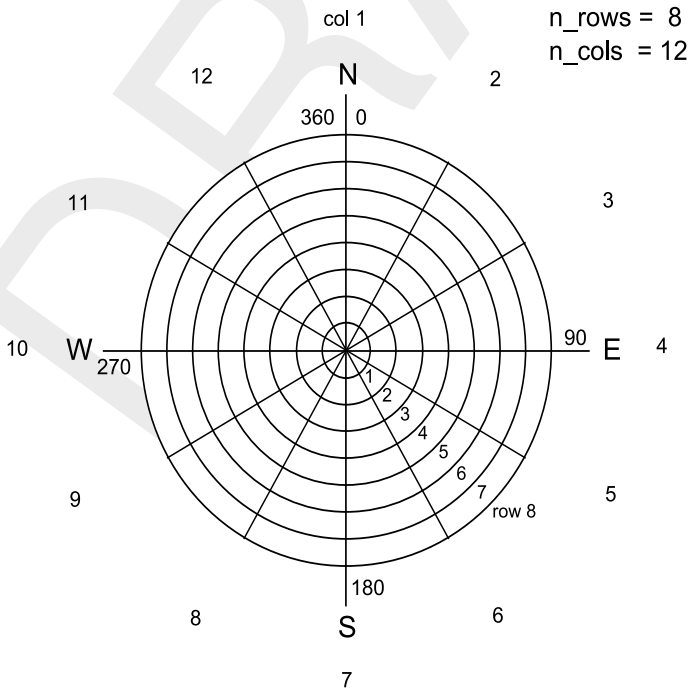

Figure A.5: *Spiderweb grid definition*

DRAFT

File version and conversion

The current description holds for `FileVersion` 1.03. The table below shows the latest modifications in the file format (and version number).

FileVersion	Modifications
`1.03`	No changes for this meteo input type
`1.02`	Changed the use of keyword `n_rows` and `n_cols`. The radius of the cyclone is divided in *n_rows* rings of width: spw_radius/n_rows [m] and the circle is divided in *n_cols* parts of $2\pi/n\_cols$ [rad].
`1.01`	Changed keyword `MeteoType` to `FileType` Changed fixed value of input type (Keyword `Filetype`) from *Spiderweb* to *meteo_on_spiderweb_grid*

Restriction:

- ◇ The restrictions for space varying wind and pressure on a Spiderweb grid are the same as for space varying wind and pressure on an equidistant grid, described in section A.2.10.2.

Remarks:

- ◇ The remarks for space varying wind and pressure on a separate curvilinear grid are the same as for space varying wind and pressure on an equidistant grid, described in section A.2.10.2.
- ◇ The Spiderweb grid is circular and the definitions of the number of rows `n_rows` and the number of columns `n_cols` is therefore different then for the other meteo input formats. For the Spiderweb grid, the number of rows determines the grid size in radial direction. The number of columns defines the grid size in angular direction. See Figure A.5.
- ◇ The wind is specified according to the nautical convention, i.e. wind from the true North has direction zero and the wind turns clockwise with an increasing angle. See Figure A.4.

Example:

A file for input of space varying wind and pressure on a 5x3 Spiderweb grid, has the following layout:

```
FileVersion            = 1.03
filetype               = meteo_on_spiderweb_grid
NODATA_value           = -999.000
n_cols                 = 3
n_rows                 = 5
grid_unit              = degree
spw_radius             = 600000.0
spw_rad_unit           = m
air_pressure_reference = air_pressure_default_from_computational_engine
n_quantity             = 3
quantity1              = wind_speed
quantity2              = wind_from_direction
quantity3              = p_drop
unit1                  = m s-1
unit2                  = degree
unit3                  = Pa
TIME          =    0.0 hours since 1997-07-14 03:00:00 -06:00
```

DRAFT

```
x_spw_eye     = 115.1
y_spw_eye     = 18.9
pdrop_spw_eye = 5300.0
1.38999       1.38261       1.38315
1.28251       1.34931       1.22571
1.27215       1.31214       1.32451
1.38999       1.86592       2.87732
1.43899       1.24912       2.21519
60.0000       180.0000      270.0000
28.7500       20.0000       31.2500
42.5000       53.7500       65.0000
49.3400       60.2400       81.5200
51.4100       62.0000       43.1200
5301.280      5294.490      5156.240
5043.460      5112.040      5264.020
5140.020      5202.520      5411.210
5294.730      5285.760      5235.250
5242.530      5156.190      5124.240
TIME          =      1.0 hours since 1997-07-14 03:00:00 -06:00
x_spw_eye     = 114.8
y_spw_eye     = 18.8
pdrop_spw_eye = 5250.0
1.35763       1.35763       1.35763
1.35763       1.87273       2.24784
1.92214       2.47836       2.17266
1.87662       2.72116       2.82375
1.26585       2.24146       2.38722
159.0000      346.5200      290.6400
342.3200      282.1400      20.2400
10.7500       25.5300       36.4500
61.8400       81.6200       45.5100
49.5250       56.7500       75.1300
5314.520      5104.490      5287.240
5124.240      5285.760      5252.420
5152.460      5247.040      5222.020
5242.020      5223.520      5475.210
5244.270      5211.210      4998.110
```

This results in the following set of meteo data. Velocities given in [m/s] and pressure drops in [Pa] on a Spiderweb grid which is given in spherical coordinates (`grid_unit` = degree). The cyclone and spiderweb grid have a radius of 600 km. The grid is 5x3, which means the radius is divided in five parts of 120 km and the 360 degrees are divided in 3 parts of 120 degrees each. Wind speeds, wind directions and pressure drops are given at two times: 0 and 1.0 hour since July 14th, 1997, 03:00 AM, in UTC-6. Between these two times the cyclone eye moves from (longitude, latitude) (115.1, 18.9) to (114.8, 18.8) on the globe and the pressure drop in the cylcone eye decreases from 5300.0 [Pa] to 5250.0 [Pa].

DRAFT

B Definition of SWAN wave variables

In SWAN a number of variables, mostly related to waves are used in input and output. The definitions of these variables are conventional for the most part.

HSIGN — significant wave height (H_s in [m]), defined as:

$$H_s = 4\sqrt{\iint E(\omega,\theta)\; d\omega d\theta}$$

where $E(\omega,\theta)$ is the variance density spectrum

TM01 — mean absolute wave period (in s) of $E(\omega,\theta)$, defined as:

$$T_{m01} = 2\pi \left(\frac{\iint \omega E(\sigma,\theta)\; d\sigma d\theta}{\iint E(\sigma,\theta)\; d\sigma d\theta} \right)^{-1} = 2\pi \left(\frac{\iint \omega E(\sigma,\theta)\; d\omega d\theta}{\iint E(\sigma,\theta)\; d\omega d\theta} \right)^{-1}$$

where ω is the absolute radian frequency, determined by the Doppler shifted dispersion relation.

DIR — mean wave direction (in $^\circ$, Cartesian or Nautical convention), as conventionally defined (Kuik *et al.*, 1988).

$$[DIR] = \arctan\left(\frac{\int \sin(\theta) E(\sigma,\theta)\; d\sigma d\theta}{\int \cos(\theta) E(\sigma,\theta)\; d\sigma d\theta} \right)$$

RTP — relative peak period (in s) of $E(\sigma)$ (equal to absolute peak period in the absence of currents)

DSPR — the one-sided directional width of the spectrum (directional spreading or directional standard deviation, in 0), defined as:

$$DSPR^2 = \left(\frac{180}{\pi}\right)^2 \int_0^{2\pi} \left\{ 2\sin\left(\frac{\theta - \bar{\theta}}{2} \right) \right\}^2 D(\theta)\; d\theta$$

and computed as conventionally for pitch-and-roll buoy data (Kuik *et al.* (1988); this is the standard definition for WAVEC buoys integrated over all frequencies):

$$\left(DSPR\,\frac{\pi}{180}\right)^2 = 2\left\{ 1 - \left(\left(\int \sin(\theta) \frac{\int E(\sigma,\theta)\; d\sigma}{\int E(\sigma)\; d\sigma} d\theta \right)^2 + \left(\int \cos(\theta) \frac{\int E(\sigma,\theta)\; d\sigma}{\int E(\sigma)\; d\sigma} d\theta \right)^2 \right)^{1/2} \right\}$$

MS — As input to SWAN in the commands BOUNDPAR and BOUNDSPEC the directional distribution of incident wave energy is: $D(\theta) = A\{\cos(\theta)\}^{[MS]}$ at all frequencies. [MS] is not necessarily an integer number.
[MS] is, for this directional distribution, related to the one-sided directional spread of the waves (DSPR) as follows:

DRAFT

[MS]	dspr (in °)
1.	37.5
2.	31.5
3.	27.6
4.	24.9
5.	22.9
6.	21.2
7.	19.9
8.	18.8
9.	17.9
10.	17.1
15.	14.2
20.	12.4
30.	10.2
40.	8.9
50.	8.0
60.	7.3
70.	6.8
80.	6.4
90.	6.0
100.	5.7
200	4.0
400	2.9
800	2.0

DISSIP — energy dissipation per unit time due to the sum of bottom friction, whitecapping and depth induced wave breaking (in W/m^2 of m^2/s, depending on command SET)

WLEN — the mean wavelength,

$$WLEN = 2\pi \left(\frac{\int k^p E(\sigma,\theta)\ d\sigma d\theta}{\int k^{p-1} E(\sigma,\theta)\ d\sigma d\theta} \right)^{-1}$$

see command QUANTITY (where $p = 1$ is default)

STEEPNESS — wave steepness, computed as:

$$\text{STEEPNESS} = \frac{\text{HSIGN}}{\text{WLEN}}$$

Qb — fraction of breakers [-] in expression of Battjes and Janssen (1978), see section 2.1.

TRANSP — energy transport with components $P_x = \iint \rho g c_x E(\sigma,\theta)\ d\sigma d\theta$ and $P_y = \iint \rho g c_y E(\sigma,\theta)\ d\sigma d\theta$ with x and y of the problem co-ordinate system, except in the case of output with `BLOCK` command in combination with command `FRAME`, where x and y relate to the x-axis and y-axis of the output frame.

VEL — current velocity with components in x and y direction of the problem co-ordinate system, except in the case of output with `BLOCK` command in combination with command FRAME, where x and y relate to the x-axis and y-axis of the output frame.

FORCE — wave induced force per unit surface area (gradient of the radiation stresses) with x and y of the problem co-ordinate system, except in

DRAFT

the case of output with BLOCK command in combination with command FRAME, where x and y relate to the x-axis and y-axis of the output frame.

$$F_x = -\frac{\partial S_{xx}}{\partial x} - \frac{\partial S_{xy}}{\partial y}, \quad \text{and} \quad F_y = -\frac{\partial S_{yx}}{\partial x} - \frac{\partial S_{yy}}{\partial y}$$

where S is the radiation stress tensor:

$$S_{xx} = \rho g \int \left(n \cos^2 \theta + n - \frac{1}{2} \right) E \; d\sigma d\theta$$

$$S_{xy} = S_{yx} = \rho g \int n \sin\theta \cos\theta E \; d\sigma d\theta$$

$$S_{yy} = \rho g \int \left(n \sin^2 \theta + n - \frac{1}{2} \right) E \; d\sigma d\theta$$

and n is the ratio of group velocity over phase velocity.

URMS — root-mean-square value of the orbital motion near the bottom

UBOT — root-mean-square value of the maximum of the orbital motion near the bottom $U_{bot} = \sqrt{2} U_{rms}$

LEAK — numerical loss of energy equal to $c_\theta E(\omega, \theta)$ across boundaries $\theta_1 =$[dir1] and $\theta_2 =$[dir2] of a directional sector (see command CGRID)

SETUP — the elevation of mean water level (relative to still water level) induced by the gradient of the radiation stresses of the waves

TPS — Smoothed Peak wave period. This value is obtained as the maximum of a parabolic fitting through the highest bin and two bins on either side of the highest one of the discrete wave spectrum. This 'non-discrete' or 'smoothed' value is a better estimate of the 'real' peak period compared to the quantity RTP.

Cartesian direction convention: the direction is the angle between the vector and the positive x-axis, measured counter-clockwise (the direction where the waves are going to or where the wind is blowing to).

Nautical direction convention: the direction of the vector from geographic North measured clockwise + 180° (the direction where the waves are coming from or where the wind is blowing from).

DRAFT

DRAFT

C Example of MDW-file Siu-Lam

In this appendix the MDW-file for the Siu Lam case is provided <siu.mdw>. Generated by the WAVE-GUI 4.94.00:

```
[WaveFileInformation]
   FileVersion          = 02.00
[General]
   ProjectName          = Siu-Lam
   ProjectNr            = 001
   Description          = Tutorial Delft3D-WAVE
   Description          = Siu Lam model
   Description          = SWAN wave model using a curvilinear grid
   OnlyInputVerify      = false
   SimMode              = stationary
   DirConvention        = nautical
   ReferenceDate        = 2005-10-01
   ObstacleFile         = siu_lam_obstacles.obs
   WindSpeed            =  2.0000000e+001
   WindDir              =  2.5500000e+002
[TimePoint]
   Time                 =  1.0800000e+003
   WaterLevel           = -1.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
[TimePoint]
   Time                 =  1.2600000e+003
   WaterLevel           =  0.0000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
[TimePoint]
   Time                 =  1.4400000e+003
   WaterLevel           =  1.5000000e+000
   XVeloc               =  0.0000000e+000
   YVeloc               =  0.0000000e+000
[Constants]
   WaterLevelCorrection =  0.0000000e+000
   Gravity              =  9.8100004e+000
   WaterDensity         =  1.0250000e+003
   NorthDir             =  9.0000000e+001
   MinimumDepth         =  5.0000001e-002
[Processes]
   GenModePhys          = 3
   WaveSetup            = false
   Breaking             = true
   BreakAlpha           =  1.0000000e+000
   BreakGamma           =  7.3000002e-001
   Triads               = false
   TriadsAlpha          =  1.0000000e-001
   TriadsBeta           =  2.2000000e+000
   BedFriction          = jonswap
   BedFricCoef          =  6.7000002e-002
   Diffraction          = false
   DiffracCoef          =  2.0000000e-001
   DiffracSteps         = 5
   DiffracProp          = true
   WindGrowth           = true
   WhiteCapping         = Komen
   Quadruplets          = true
   Refraction           = true
   FreqShift            = true
   WaveForces           = dissipation
[Numerics]
   DirSpaceCDD          =  5.0000000e-001
   FreqSpaceCSS         =  5.0000000e-001
```

DRAFT

```
   RChHsTm01            =  2.0000000e-002
   RChMeanHs            =  2.0000000e-002
   RChMeanTm01          =  2.0000000e-002
   PercWet              =  9.8000000e+001
   MaxIter              = 4
[Output]
   TestOutputLevel      = 0
   TraceCalls           = false
   UseHotFile           = false
   WriteCOM             = false
   LocationFile         = siu.loc
   WriteTable           = true
   WriteSpec1D          = true
   WriteSpec2D          = true
[Domain]
   Grid                 = siu_lam.grd
   BedLevel             = siu_lam.dep
   DirSpace             = circle
   NDir                 = 36
   StartDir             =  0.0000000e+000
   EndDir               =  0.0000000e+000
   FreqMin              =  5.0000001e-002
   FreqMax              =  1.0000000e+000
   NFreq                = 24
   Output               = true
[Boundary]
   Name                 = Boundary 1
   Definition           = orientation
   Orientation          = west
   SpectrumSpec         = parametric
   SpShapeType          = gauss
   PeriodType           = peak
   DirSpreadType        = degrees
   PeakEnhanceFac       =  3.3000000e+000
   GaussSpread          =  3.3000000e+000
   DistanceDir          = counter-clockwise
   CondSpecAtDist       =  1.5000000e+003
   WaveHeight           =  0.0000000e+000
   Period               =  5.0000000e+000
   Direction            =  2.5500000e+002
   DirSpreading         =  4.0000000e+000
   CondSpecAtDist       =  9.0000000e+003
   WaveHeight           =  1.0000000e+000
   Period               =  5.0000000e+000
   Direction            =  2.5500000e+002
   DirSpreading         =  4.0000000e+000
```

DRAFT

D DATSEL data extraction utility

D.1 Function

DATSEL is used to select data from a NEFIS map-file. It produces an ASCII datafile in TEKAL format.

D.2 Running DATSEL

Follow the instructions in Chapter 3 to get to the **Waves** selection window, see Figure 3.2.

◇ Select *Tools* in the **Waves (standalone)** selection window, next Figure D.1 is displayed.

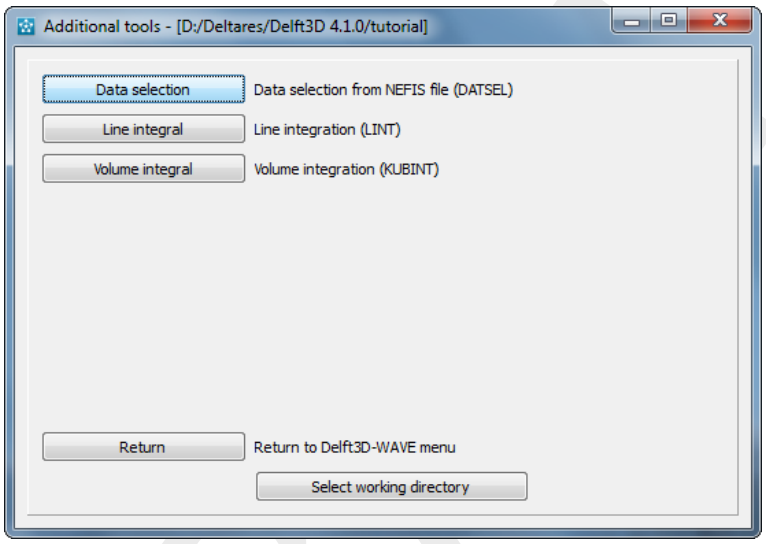

Figure D.1: *Selection window for Waves Tools*

◇ Select *Data selection* to start DATSEL.

The program then asks for an input filename. Enter just the filename if the input file is in the current directory, or the full path/filename if it is somewhere else. If you do not specify a file, but just press enter, the program will interactively ask for the input items specified in the following section.

D.3 Input description

Record 1 Filetype number (1-6).

Number	Filetype
1	Communication file (com-)
2	Transport map file (tram-)
3	Flow map file (trim-)
4	Bottom map file (botm-)
5	Waves (HISWA) output result file (hwgxy-)

6	Waves output bottom file (bagr-)
7	Waves (SWAN) output result file (wavm-)

Record 2 Function number

Filetype = 1	
1	initial bed level
2	time-varying bed level
3	water level
4	Hrms wave height
5	Hrms wave vector
6	Tp wave period
7	wave dissipation
8	velocity
9	discharge
10	wave force
11	mass flux
12	tide-averaged bedload transport
13	tide-averaged suspended transport
14	maximal bottom friction
Filetype = 2	
1	tide-averaged bedload transport
2	tide-averaged suspended transport
3	bed-load transport
4	suspended load transport
Filetype = 3	
1	initial bottom depth
2	depth water level points
3	water level
4	velocity
5	bottom stress
6	thickness of bed layer
7	time-varying depth
8	bottom sediment kg/m^2
9	bed load transport
10	suspended load transport
11	constituent
12	averaged bed-load transport
13	averaged suspended transport
Filetype = 4	
1	time-integrated transport
Filetype = 5	

DRAFT

1	Hsig wave height
2	Hsig wave vector
3	wave period
4	directional spreading
5	dissipation
6	leakage
7	fraction breaking
8	orbital velocity
9	wave steepness
10	wave length
11	current velocity
12	energy transport
Filetype = 6	
1	contract dredging depth
2	cumulative dredging depth
3	bed level
Filetype = 7	
1	Hsig wave height
2	Hsig wave vector
3	wave period
4	directional spreading
5	dissipation
6	leakage
7	fraction breaking
8	orbital velocity
9	wave steepness
10	wave length
11	current velocity
12	energy transport
13	peak period

Record 3	Number of time steps
	For each time step i
Record 4r	Time step number i (one per record)
Record 5	Time-varying output (type 1) or time-average output (type 2)
Record 6	Path name (including last \)
Record 7	Case (3 characters).
Record 8	Label (max. 4 characters).
Record 9	Output filename.

D.4 Output files

A file called <datsel.log> will be created in the working directory. A TEKAL datafile will be created with the name given by you.

DRAFT

D.5 Example file

Based on the following input file, the program computes the time-average of the first three bed levels on the communication file <d:\delft3d\com-xp1a.dat> (and *.def) and writes the result to a text file: <d:\output\avgbed.txt>.

File contents:	Explanation (not part of the file)
`1`	communication file
`2`	time-varying bed level
`3`	3 time steps
`1`	time step 1
`2`	time step 2
`3`	time step 3
`2`	time-average output
`d:\delft3d\`	working directory, ending with \
`xp1`	case name
`a`	label, communication file used: com-xp1a.dat (and .def)
`d:\output\avgbed.txt`	output file name

DRAFT

E LINT Line Integration

E.1 Function

LINT (for Line INTegral) computes the line integral of a 2D vector quantity over specified polylines. It produces detailed results along the polylines and integrated results over each polyline. The polylines can be defined with RGFGRID or QUICKIN.

E.2 Running LINT

Follow the instructions in Chapter 3 to get to the **Waves** selection window, see Figure 3.2.

◇ Select *Tools* in the **Waves (standalone)** selection window, next Figure E.1 is displayed.

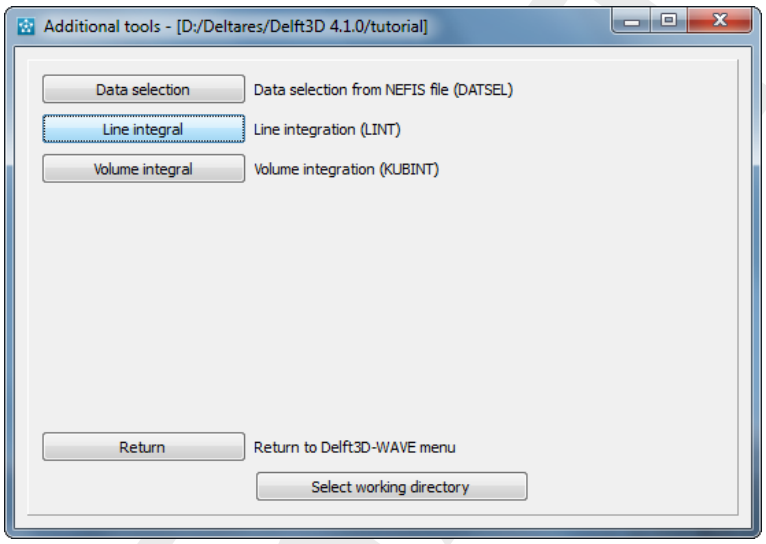

Figure E.1: *Selection window for Waves Tools*

◇ Select *Linear integral* to start LINT.

The program then asks for an input filename. Enter just the filename if the input file is in the current directory, or the full path/filename if it is somewhere else. If you do not specify a file, but just press enter, the program will interactively ask for the input items specified in the following section.

E.3 Input description

Record 1	Filename TEKAL datafile (e.g. obtained from DATSEL)
Record 2	Column numbers x, y, u, v
Record 3	Filename detailed output
Record 4	Filename integrated output
Record 5	Number of subdivisions per polygon element
Record 6	Detailed screen output (0/1 i.e. no/yes)
Record 7	Filename with polylines (e.g. obtained from RGFGRID or QUICKIN)

For LINT versions older than 2.00.00:

Record 7	Number of polylines
	For each polyline i
Record 8r	Number of points polyline i

	For each point
Record 9r	x, y point j

Remarks:

- ◇ The maximum number of points per polyline is 100
- ◇ The total number of subdivisions per polyline should be less than 10.000.
- ◇ The integration is based on the specified number of equidistant subdivisions per polygon element.
- ◇ The results from DATSEL are given at the water level points. Hence, the grid constructed by LINT has the water level points as corner points.

E.4 Output files

A log file called <lint.log> is produced in the working directory. Detailed and integrated output are written to the files with specified names.

E.5 Example file

Based on the following input file, the program interpolates the data stored in the TEKAL file <d:\data\sedtr.txt> (first column: x co-ordinates, second column: y co-ordinates, third and fourth columns: transport in x and y directions) to 100 points: 50 points between (0,0) and (100,0) and 50 between (100,0) and (100,100). The output is written to <d:\data\detout.txt>. The integrated transport is computed and written to <d:\data\intgout.txt>.

LINT version 2.00.00 or higher:

File contents:	Explanation (not part of the file)
`d:\data\sedtr.txt`	TEKAL input file
`1 2 3 4`	column numbers for x, y, u, v
`d:\data\detout.txt`	output file for detail information
`d:\data\intgout.txt`	output file for integrated data
`50`	subdivisions per polyline element
`0`	no detailed screen output
`lint.pol`	filename with polyline

File <lint.pol> may look like:

```
BL01
  3    2
0.0   0.0
100.0   0.0
100.0   100.0
```

LINT version older than 2.00.00:

DRAFT

File contents:	Explanation (not part of the file)
`d:\data\sedtr.txt`	TEKAL input file
`1 2 3 4`	column numbers for x,y,u,v
`d:\data\detout.txt`	output file for detail information
`d:\data\intgout.txt`	output file for integrated data
`50`	subdivisions per polyline element
`0`	no detailed screen output
`1`	one polyline
`3`	three points (two polyline elements)
`0.0 0.0`	x, y co-ordinates of first point
`100.0 0.0`	x, y co-ordinates of second point
`100.0 100.0`	x, y co-ordinates of third point

DRAFT

DRAFT

F KUBINT volume integration

F.1 Function

KUBINT computes the integral of a 2D function over the areas enclosed by specified polygons. The polygons can be defined with RGFGRID or QUICKIN.

F.2 Running KUBINT

Follow the instructions in Chapter 3 to get to the **Waves** selection window, see Figure 3.2.

◇ Select *Tools* in the **Waves (standalone)** selection window, next Figure F.1 is displayed.

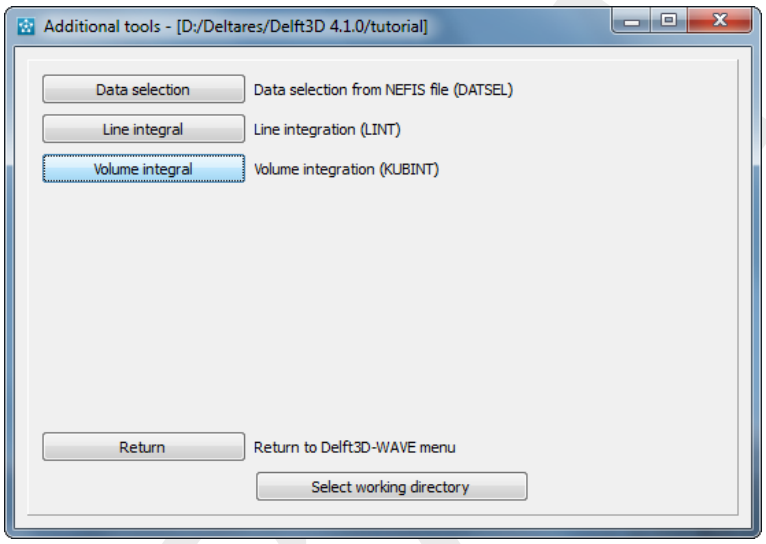

Figure F.1: *Selection window for Morphology Tools*

◇ Select *Volume integral* to start KUBINT.

The program then asks for an input filename. Enter just the filename if the input file is in the current directory, or the full path/filename if it is somewhere else. If you do not specify a file, but just press enter, the program will interactively ask for the input items specified in the following section.

F.3 Input description

Record 1	Filename TEKAL datafile (e.g. obtained from DATSEL)
Record 2	Column numbers x, y, function
Record 3	Filename output
Record 4	Number of pixels in x- and y-direction
Record 5	Detailed screen output (0/1 i.e. no/yes)
Record 6	Filename with polylines (e.g. obtained from RGFGRID or QUICKIN)

For KUBINT versions older than 2.00.00:

Record 6	Number of polygons
	For each polygon i
Record 7r	Number of points polygon i
	For each point
Record 8r	x, y point j

Remarks:

- ◇ The maximum number of polygons is 500.
- ◇ The maximum number of points per polygon is 1000.
- ◇ The total number of pixels (square of number specified in Record 4) should be less than 250.000.
- ◇ The integration is based on the specified number of pixels in x (resp. y) direction stretched to fill the distance between min(x) and max(y). The pixels are generally non-square.
- ◇ The results from DATSEL are given at the water level points. Hence, the grid constructed by KUBINT has the water level points as corner points.

F.4 Output files

A log file called <kubint.log> is produced in the working directory. Output is written to the file with specified name.

F.5 Example file

Based on the following input file, the program interpolates the data stored in the TEKAL file <d:\data\bedlvl.txt> (first column: x co-ordinates, second column: y co-ordinates, third column: bed level) onto rectangular grid with 502 points: 50 points uniform in x direction and 50 points uniform in y direction. The results are integrated and the end result is written to <d:\data\volint.txt>.

KUBINT version 2.00.00 or higher:

File contents:	Explanation (not part of the file)
`d:\data\bedlvl.txt`	TEKAL input file
`1 2 3`	column numbers for x, y, val
`d:\data\volint.txt`	output file
`50`	number of pixels in x- and y- direction
`0`	no detailed screen output
`kubint.pol`	filename with polyline

File <kubint.pol> may look like:

```
BL01
  3    2
0.0   0.0
100.0   0.0
100.0   100.0
```

KUBINT version older than 2.00.00:

DRAFT

File contents:	Explanation (not part of the file)
`d:\data\bedlvl.txt`	TEKAL input file
`1 2 3`	column numbers for x, y, val
`d:\data\volint.txt`	output file
`50`	number of pixels in x and y direction
`0`	no detailed screen output
`1`	one polyline
`3`	three points
`0.0 0.0`	x, y co-ordinates of first point
`100.0 0.0`	x, y co-ordinates of second point
`100.0 100.0`	x, y co-ordinates of third point

DRAFT

DRAFT

DRAFT

DRAFT

Deltares systems

PO Box 177
2600 MH Delft
Rotterdamseweg 185
2629 HD Delft
The Netherlands

+31 (0)88 335 81 88
sales@deltaressystems.nl
www.deltaressystems.nl